



---

***SISTEMAS DE INFORMAÇÃO GEOGRÁFICA  
E VIDEOJOGOS***

---

Rui Miguel Bento Guerreiro

---

Dissertação apresentada como requisito parcial para obtenção do  
grau de Mestre em Ciência e Sistemas de Informação Geográfica

---

Instituto Superior de Estatística e Gestão de Informação  
da Universidade Nova de Lisboa

# **SISTEMAS DE INFORMAÇÃO GEOGRÁFICA E VIDEJOGOS**

Dissertação orientada por

Professor Doutor Marco Octávio Trindade Painho

Novembro de 2014

## AGRADECIMENTOS

Gostaria de expressar o meu agradecimento a todos os colegas, professores e amigos sem os quais a realização deste trabalho não teria sido possível:

Em primeiro lugar, ao Professor Doutor Marco Painho pela sua disponibilidade, orientação e correcção deste trabalho.

Aos meus ex-colegas e amigos Sérgio Godinho, Rui Machado e Catarina Rodrigues, pela boa disposição e honestidade com que criticaram o trabalho nas várias fases de desenvolvimento.

A todos aqueles que, mesmo sem uma relação directa com o trabalho, tiveram uma influência positiva, em particular aos meus grandes amigos Desidério Lázaro, Pedro Viana e Tiago Garcia, à Professora Doutora Isabel Ramos, e aos meus actuais colegas Renata Salada, Luís Pontes, Diana Paiva, Tiago Henriques, Rita Brito e Denise Paulino pela compreensão, tolerância e incentivo.

Em último lugar gostaria de agradecer às minhas Avós, aos meus Pais e à minha Irmã pelo apoio e paciência, e ao meu filho David por todos os sorrisos.

A minha mais profunda gratidão vai para a minha cara-metade, Estela de Jesus, que esteve ao meu lado desde o dia 0, resistiu aos meus piores momentos, nunca deixou de acreditar e me carregou nos ombros sempre que me falharam as forças. Sem ti não tinha conseguido.

# SISTEMAS DE INFORMAÇÃO GEOGRÁFICA E VIDEJOGOS

## RESUMO

O espaço é um elemento fundamental no *design* de videojogos. A tecnologia actual dá ao jogador a possibilidade de experienciar mundos de grandes dimensões, sejam eles puramente imaginários, ou réplicas de ambientes reais. As exigências da produção destes mundos fazem com que os métodos de criação manuais não sejam suficientes, e haja recurso a ferramentas de geração procedimental de conteúdos. No que diz respeito aos mapas, estes procedimentos assemelham-se aos processos de modelação geográfica utilizados nos Sistemas de Informação Geográfica, mas são raros os casos em que há a utilização conjunta das tecnologias. Este trabalho replica métodos de geração procedimental de mapas em videojogos com recurso a *software* de Sistemas de Informação Geográfica, indentifica as mais-valias desta abordagem e aponta duas alternativas, cujo desenvolvimento pode contribuir para a aproximação das duas áreas.

# SISTEMAS DE INFORMAÇÃO GEOGRÁFICA E VIDEJOGOS

## ABSTRACT

Space is a fundamental element in videogame design. Current technology allows the player to experience large open worlds, purely imaginary or real landscapes' replicas. The manual creation of these large open worlds isn't enough to satisfy the production demands, meaning that tools for the procedural content generation must be used. Regarding maps, these procedures are very similar to the modeling processes that we see in Geographical Information Systems, but very rarely do we see the combined usage of both technologies. This work replicates procedural content generation of maps for videogames using Geographical Information *Software*, identifies the positive aspects of this approach and points out two alternatives, which development can contribute to bring both areas together.

## **PALAVRAS CHAVE**

Sistemas de Informação Geográfica

Videojogos

Geração Procedimental de Conteúdos

Análise Espacial

Fractais

Ruído Coerente

Modelação Digital do Terreno

## **KEYWORDS**

Geographic Information Systems

Videogames

Procedural Content Generation

Spatial Analysis

Fractals

Coherent Noise

Digital Terrain Modelling

## ACRÓNIMOS

**ASCII:** *American Standard Code for Information Interchange*

**ESRI:** *Environmental Systems Research Institute*

**fBm:** movimento fraccional Browniano (do inglês *fractional Brownian Motion*)

**MDT:** Modelo Digital do Terreno

**MS&T:** Simulação e Treino Militar (do inglês *Military Simulation and Training*)

**PCG:** Geração Procedimental de Conteúdos (do inglês *Procedural Content Generation*)

**RPG:** *Role Playing Game*

**SDK:** *Software Development Kit*

**SIG:** Sistemas de Informação Geográfica

**TIN:** *Triangulated Irregular Network*

## ÍNDICE DO TEXTO

<b>Agradecimentos .....</b>	<b>iii</b>
<b>Resumo .....</b>	<b>iv</b>
<b>Abstract.....</b>	<b>v</b>
<b>Palavras Chave .....</b>	<b>vi</b>
<b>Keywords .....</b>	<b>vi</b>
<b>Acrónimos .....</b>	<b>vii</b>
<b>Índice de Tabelas.....</b>	<b>xi</b>
<b>Índice de Figuras.....</b>	<b>xii</b>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 ENQUADRAMENTO.....	1
1.2 OBJECTIVOS.....	2
1.3 ESTRUTURA DA TESE.....	2
<b>2 ENQUADRAMENTO TEÓRICO.....</b>	<b>4</b>
2.1 LUDOLOGIA .....	4
2.1.1 <i>Definição de Jogo</i> .....	4
2.1.2 <i>A experiência do jogador</i> .....	5
2.1.3 <i>Os jogos como narrativa</i> .....	11
2.1.4 <i>O papel do Espaço no Design de um Jogo</i> .....	13
2.1.5 <i>Mapas em Videojogos</i> .....	14
2.2 GERAÇÃO PROCEDIMENTAL DE CONTEÚDOS .....	17
2.2.1 <i>Dimensões da geração procedimental</i> .....	19
2.2.2 <i>Classificação e taxonomia dos tipos de conteúdos e métodos de geração</i> .....	21
2.2.3 <i>Espaço de jogo:</i> .....	23
2.2.4 <i>Geradores de números pseudo aleatórios</i> .....	25
2.2.4.1 <i>Ruído Perlin</i> .....	27
2.2.5 <i>Fractais</i> .....	30
2.2.5.1 <i>Movimento Fraccional Browniano</i> .....	30
2.2.5.2 <i>Deslocamento do ponto central</i> .....	32
2.3 SISTEMAS DE INFORMAÇÃO GEOGRÁFICA .....	34
2.3.1 <i>Modelos de Dados</i> .....	35
2.3.2 <i>Modelos Digitais do Terreno</i> .....	37



2.3.3 <i>Análise espacial</i> .....	40
2.4 SISTEMAS DE INFORMAÇÃO GEOGRÁFICA E VIDEOJOGOS .....	43
<b>3 METODOLOGIA</b> .....	<b>46</b>
3.1 FERRAMENTAS .....	47
3.1.1 <i>Motor de Jogo</i> .....	47
3.1.2 <i>Aplicação Desktop de SIG</i> .....	47
3.1.3 <i>Extensões Python</i> .....	48
3.2 PROCEDIMENTOS .....	50
3.2.1 <i>Primitivas</i> .....	50
3.2.1.1 <i>Ruído Perlin (fBm)</i> .....	51
3.2.1.2 <i>Deslocamento do ponto central</i> .....	52
3.2.2 <i>Modelo Geral de Aplicação</i> .....	54
3.2.3 <i>Altimetria</i> .....	55
3.2.4 <i>Rede Hidrográfica</i> .....	56
3.2.5 <i>Temperatura</i> .....	57
3.2.6 <i>Humidade</i> .....	57
3.2.7 <i>Biomass</i> .....	58
3.2.8 <i>Importação para o motor de jogo</i> .....	60
<b>4 RESULTADOS</b> .....	<b>61</b>
4.1 PRIMITIVAS .....	61
4.2 ALTIMETRIA .....	62
4.3 REDE HIDROGRÁFICA .....	63
4.4 TEMPERATURA .....	63
4.5 HUMIDADE .....	64
4.6 BIOMASS .....	65
4.7 MOTOR DE JOGO .....	65
<b>5 DISCUSSÃO DE RESULTADOS</b> .....	<b>67</b>
5.1 PRIMITIVAS .....	67
5.2 ALTIMETRIA .....	69
5.3 REDE HIDROGRÁFICA .....	70
5.4 TEMPERATURA .....	71
5.5 HUMIDADE .....	72
5.6 BIOMASS .....	73
5.7 MOTOR DE JOGO .....	74
5.8 CONSIDERAÇÕES GERAIS .....	75

<b>6 CONCLUSÕES .....</b>	<b>76</b>
<b>7 BIBLIOGRAFIA .....</b>	<b>78</b>
<b>ANEXO 1 – CÓDIGO PYTHON DO SCRIPT PERLIN FBM.....</b>	<b>85</b>
<b>ANEXO 2 – CÓDIGO PYTHON DO SCRIPT DIAMOND SQUARE .....</b>	<b>86</b>

## ÍNDICE DE TABELAS

Tabela 1: Classificação quanto ao espaço de alguns jogos de aventura/RPG. ....	14
Tabela 2: Classificação dos tipos de conteúdos que podem ser gerados proceduralmente (HENDRIKX et al. 2013) .....	22
Tabela 3: Taxonomia de métodos comuns de geração de conteúdos para jogos (HENDRIKX et al., 2013) .....	23
Tabela 4: Características diferenciadoras de SIG e Videojogos. Adaptado de PAGULAYAN (2003) .....	43
Tabela 5: Descrição dos parâmetros do cabeçalho de um ficheiro ASCII Raster (ESRI, 2012) .....	49
Tabela 6: Matriz da adaptação do diagrama de WHITTAKER (1975). ....	58
Tabela 7: Parâmetros dos scripts para geração das primitivas. ....	61
Tabela 8: Valor de corte da operação Spatial Analyst Tools > Conditional > Con. ....	62
Tabela 9: Acumulação do escoamento - valor de corte da operação Spatial Analyst Tools >Conditional > Con. ....	63
Tabela 10: Parâmetros do script para geração do ruído da temperatura. ....	64
Tabela 11: Parâmetros dos scripts para geração do ruído da humidade. ....	64

## ÍNDICE DE FIGURAS

Figura 1: Modelo generalizado da experiência de jogo. Adaptado de SUTTON-SMITH (1986) .....	6
Figura 2: Canal do flow (MARCZEWSKI, 2012).....	8
Figura 3: A viagem do jogador (KIM, 2012) .....	9
Figura 4: Flow e a viagem do jogador (MARCZEWSKI, 2012) .....	10
Figura 5: Mapa Mundo no jogo Final Fantasy VII. Fonte: CONCELMO (2009) .....	16
Figura 6: Mapa gerado através do algoritmo de deslocamento do ponto central (Fonte: <a href="http://www.avanderw.co.za/">http://www.avanderw.co.za/</a> ) .....	24
Figura 7: Exemplo de um mapa do jogo Minecraft (fonte: <a href="http://www.minecraftforum.net">www.minecraftforum.net</a> ) .....	24
Figura 8: Biomas. Adaptado de WHITTAKER (1975). ....	25
Figura 9: Ruído Branco.....	26
Figura 10: Ruído Coerente (ruído Perlin) .....	27
Figura 11: Pormenor de uma quadrícula da grelha de gradientes. ....	28
Figura 12: Vectores da posição de $x$ relativamente aos vértices da grelha.....	29
Figura 13: Aspecto da função de interpolação. ....	29
Figura 14: fBm a uma dimensão, com $H$ a variar de 1.0 (mais suave) no topo, para 0.0 (mais rugoso) em intervalos de 0.2. (EBERT et al., 2003).....	31
Figura 15: fBm a duas dimensões. $H = 1.0$ (ganho 0.5), Lacunaridade = 2.0 e oitavas = 5. ....	32
Figura 16: Esquema do algoritmo diamante-quadrado. a) determinação do primeiro ponto central; b) 1º passo diamante; c) 1º passo quadrado; d) 2º passo diamante .....	33
Figura 17: Superfície gerada através do algoritmo diamante-quadrado.....	33
Figura 18: Representação de entidades pelo modelo Vectorial (à esquerda) e Raster (à direita) (ESRI, 2012). ....	36
Figura 19: Componentes da Modelação Digital do Terreno. Adaptado de Li et al (2010).....	37
Figura 20: Modelo digital do terreno: Grid à esquerda; TIN à direita.....	39
Figura 21: Fill - Remoção de picos e áreas de drenagem interna (ESRI, 2012) .....	41
Figura 22: Direcção do escoamento (ESRI, 2012) .....	42
Figura 23: Acumulação do Escoamento (ESRI, 2012).....	43

Figura 24: Localização do script Perlin fBm na toolbox e caixa de diálogo.....	51
Figura 25: Localização do script Perlin fBm na toolbox e caixa de diálogo.....	53
Figura 26: Modelo geral de aplicação.....	54
Figura 27: Esquema da geração da altimetria. ....	55
Figura 28: Esquema da geração da rede hidrográfica. ....	56
Figura 29: Esquema da geração da temperatura.....	57
Figura 30: Esquema da geração da humidade. ....	57
Figura 31: Esquema da geração de biomas. ....	59
Figura 32: Primitivas – Perlin à esquerda, deslocamento do ponto central à direita.....	62
Figura 33: Altimetrias – Perlin à esquerda, deslocamento do ponto central à direita. ....	62
Figura 34: Rede hidrográfica – Perlin à esquerda, deslocamento do ponto central à direita. ....	63
Figura 35: Temperaturas – Perlin à esquerda, deslocamento do ponto central à direita. ....	64
Figura 36: Humidades – Perlin à esquerda, deslocamento do ponto central à direita. ....	65
Figura 37: Biomas – Perlin à esquerda, deslocamento do ponto central à direita. ....	65
Figura 38: À esquerda – vista geral do modelo do terreno com distribuição de biomas; à direita – modo de jogo com vista na primeira pessoa. ....	66
Figura 39: Distribuições – Perlin fBm à esquerda; Diamond Square à direita. ....	67
Figura 40: Vestígios da aplicação do algoritmo de deslocamento do ponto central com componente aleatória no intervalo [0; 1]. ....	68
Figura 41: Variação da cobertura dos biomas face à variação do limiar de acumulação do escoamento (em função do escoamento máximo verificado). ....	70
Figura 42: Pormenor da acumulação do escoamento na altimetria Perlin. ....	71
Figura 43: Variação da cobertura dos biomas face à variação da amplitude do script de ruído (em função da distância máxima às linhas de água).....	72
Figura 44: Exemplo da humidade do deslocamento do ponto central: resultado com amplitude do ruído igual a 10% da distância máxima à esquerda; 50% à direita. ....	73

### 1.1 ENQUADRAMENTO

A tecnologia por detrás da indústria dos videojogos oferece aos jogadores a possibilidade de interagir com mundos de grandes dimensões. Alguns desses mundos correspondem a réplicas de ambientes urbanos ou paisagens reais, mas a maioria são fictícios no domínio do fantástico. Nestes casos, os mundos têm que ser criados de raiz, e o avançar da tecnologia tem permitido resultados com escalas e níveis de detalhe muito elevados.

Um levantamento dos métodos utilizados para a criação deste tipo de conteúdos permite estabelecer algum paralelismo com as técnicas de modelação geográfica e análise espacial utilizadas em Sistemas de Informação Geográfica (SIG), mas não há uma convergência estabelecida no que diz respeito à utilização conjunta de ambas as tecnologias.

Colocam-se, portanto, as seguintes questões:

- Faz sentido a utilização de SIG no *design* e construção dos mundos fictícios dos videojogos?
- Como é possível gerar um mundo fictício através de SIG?
- Quais as mais-valias da utilização de SIG neste processo?

Este trabalho pretende explorar estas questões, abordando por um lado a importância do espaço nos videojogos e por outro a capacidade dos SIG para a criação e análise desse mesmo espaço.

## 1.2 OBJECTIVOS

No seguimento do enquadramento anterior, constituem objectivos do presente trabalho:

- Identificar pontos de convergência entre as tecnologias de SIG e videojogos;
- Utilizar uma abordagem SIG para gerar um mundo fictício onde se desenrola um videojogo;
- Identificar um conjunto de mais-valias na geração procedimental utilizando SIG;
- Propor abordagens complementares.

## 1.3 ESTRUTURA DA TESE

Por forma a atingir os objectivos propostos, a tese encontra-se organizada em 6 capítulos.

No capítulo introdutório é definido o mote do trabalho, bem como os objectivos e organização do mesmo.

No enquadramento teórico serão abordadas as questões teóricas de ludologia e design de videojogos que estabelecem o espaço como um dos elementos principais na construção da experiência do jogador. Numa segunda fase, serão expostos os tipos de conteúdos que são gerados procedimentalmente e detalhados os dois métodos mais proeminentes de geração procedimental de espaços de jogo. No final serão introduzidos os SIG e identificados os pontos de convergência entre as duas tecnologias.

A metodologia centrar-se-á na integração de dois algoritmos de geração procedimental de superfícies num *software* de SIG, e utilização deste último para gerar e modelar um conjunto de variáveis que em última instância conduzem à criação de um mapa exterior para ser utilizado num videojogo.

Nos resultados serão apresentadas as superfícies geradas e os mapas exteriores finais obtidos a partir de duas superfícies iniciais distintas.

No quinto capítulo serão discutidos os pontos fortes e pontos fracos de cada um dos resultados produzidos, bem como justificadas as escolhas de parâmetros e o seu impacto no resultado final.

O último capítulo irá reflectir sobre as vantagens e desvantagens dos métodos utilizados, e sobre o cruzamento entre as aplicações de SIG e videojogos de forma a apontar melhorias desejadas em ambas as partes e sugerir novas abordagens para trabalhos futuros.



### 2.1 LUDOLOGIA

Uma exposição da teoria dos jogos (não confundir com teoria de jogo), bem como das várias escolas de pensamento sobre os mesmos seria um exercício demasiado extenso, o que acabaria por contribuir para a dissipação do tema. Por esse motivo, esta secção está delineada de forma a focar os pontos-chave necessários para compreender o que é um jogo e o papel do espaço no *design* do mesmo:

1. Definição de Jogo;
2. A experiência do jogador;
3. Os jogos como narrativa;
4. O papel do espaço nos videojogos;
5. Mapas em videojogos.

#### 2.1.1 DEFINIÇÃO DE JOGO

A definição de jogo não é consensual. SALEN e ZIMMERMAN (2004) fazem o levantamento de 8 definições distintas de jogo, antes de oferecerem a sua versão formal, baseada nas ideias principais das restantes:

*“Um jogo é um sistema em que os jogadores entram em conflito artificial, definido por regras, que produz resultados quantificáveis.”*

É uma definição é aplicável a todos os jogos quer sejam jogos de computador, jogos de tabuleiro ou de cariz desportivo.

Esta definição formal, bem como as que lhe deram origem apresentam os jogos como se existissem em mundos próprios, desligados da realidade. KOSTER (2013) refere que os jogos são puzzles a resolver tal como qualquer outra coisa na vida, e que estão ao mesmo nível de, por exemplo, aprender a guiar, tocar um instrumento musical ou

decorar a tabuada. Isto porque partilham a forma como aprendemos os padrões da realidade, trabalhamos a memória muscular e criamos os reflexos prontos a serem invocados quando necessário. A única diferença entre jogos e realidade é que a parada é muito mais baixa nos jogos.

HUIZINGA (1938), historiador considerado um dos maiores académicos da ludologia do séc. XX, refere na secção de abertura do seu livro *Homo Ludens* que:

*“Jogar é uma função significativa - quer isto dizer que tem algum sentido. No jogo, está sempre alguma coisa ‘em jogo’ que transcende as necessidades imediatas da vida, e que confere significado à acção. Todos os jogos significam alguma coisa.”*

SALEN e ZIMMERMAN (2004) consideram que esta passagem invoca aquele que é o princípio do *design* de jogos: o objectivo de um *design* de sucesso é a criação de um jogo com significado.

### **2.1.2 A EXPERIÊNCIA DO JOGADOR**

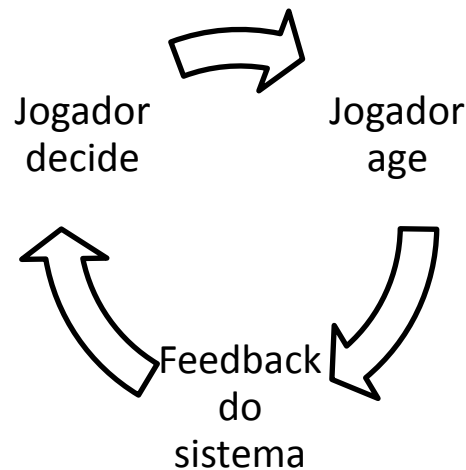
A experiência de jogo resume-se à participação. Ver um jogo de futebol da bancada não conta como jogar.

Jogar é experienciar o jogo: ver, tocar, sentir, saborear, mover o corpo, sentir, comunicar, alterar os padrões normais de pensar. Das várias formas de experienciar um videojogo, SUTTON-SMITH (1986) apresenta um modelo que lista estes processos:

- Percepção visual;
- Discriminação auditiva – percepção dos sons associados a eventos ou objectos no jogo;
- Resposta motora – acção física que o jogador exerce sobre o controlador;
- Concentração;
- Padrões perceptuais de aprendizagem – compreensão da estrutura do jogo em si.

Este modelo pode ser generalizado em 3 componentes (fig. 1):

- Input – o jogador age (resposta motora);
- Output – *feedback* do sistema (percepção visual; discriminação auditiva);
- Processos internos – o jogador decide (concentração; padrões perceptuais de aprendizagem).



---

Figura 1: Modelo generalizado da experiência de jogo. Adaptado de SUTTON-SMITH (1986).

O *design* de jogos é um problema de segunda ordem: o *designer* influencia indirectamente a experiência do jogador, pela definição directa das regras e *feedback* do sistema.

Apesar da influência indirecta que o *design* tem na experiência do jogador, não é deixado ao acaso. Há um conjunto de variáveis que podem ser trabalhadas para conduzir à produção de um jogo com significado. A esse respeito, há um conceito omnipresente na literatura do *design* de jogos: O *flow*.

O conceito de *flow* foi introduzido por CSIKSZENTMIHALYI (1990) para definir o estado de elevada concentração e imersão absoluta acompanhadas por um grande sentimento de gratificação.

Este conceito é transversal a actividades completamente distintas: a descrição do sentimento registado para um nadador de longa distância relativamente à travessia do Canal da Mancha, foi idêntica à de um jogador de xadrez durante um torneio, ou de um músico durante um concerto.

A fenomenologia do *flow* tem 8 componentes essenciais:

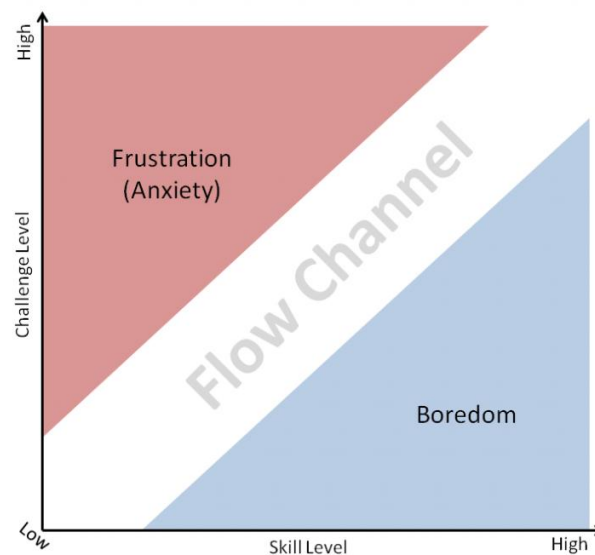
- **(1) Actividade que representa um desafio e requer perícia:** o *flow* não é experimentado de forma passiva; é necessário envolvimento activo e direccionado. Pelas palavras do próprio “a esmagadora maioria das experiências óptimas reportadas ocorrem em actividades orientadas por objectivos e com regras bem definidas”.
- **(2) Objectivos e (3) *feedback* claros:** estes dois componentes relacionam-se directamente a orientação para objectivos natural do jogos, e com a sequência acção-*feedback* discernível necessárias no processo de tomada de decisão.
- **(4) Capacidade de concentração:** um efeito comum do *flow* é a concentração total na actividade em causa, não deixando qualquer espaço para informação irrelevante, como se o *flow* criasse um espaço de experiência próprio para o sujeito.
- **(5) Fusão entre acção e percepção:** uma característica própria do *flow* é que o sujeito encontra-se tão absorvido na actividade, que todas as acções se tornam espontâneas e automáticas ao ponto de não haver distinção entre o indivíduo e a acção.
- **(6) Perda da autoconsciência:** relacionada com o pontos anteriores, num estado de *flow* o sujeito perde a noção do próprio, tornando-se parte integrante de um sistema - a “experiência” - cuja forma emerge das regras da actividade, e a energia da atenção do sujeito.
- **(7) Paradoxo do controlo:** na experiência óptima, o sujeito consegue exercer controlo absoluto, sem estar em controlo da situação. Numa actividade suficientemente difícil com resultado incerto, um sujeito sabe que tem o controlo quando mesmo nessas condições consegue influenciar o desfecho.
- **(8) Distorção do tempo:** a noção de tempo perde-se durante o estado de *flow*.

Apesar de CSIKSZENTMIHALYI (1990) não fazer esta distinção, Salen e Zimmerman (2004) separam estes componentes em 2 grupos:

1. Fusão entre acção e percepção
  - Capacidade de Concentração
  - Perda da autoconsciência
  - Distorção do tempo
2. Actividade que representa um desafio e requer perícia
  - Objectivos claros
  - Feedback* claro
  - Paradoxo do controlo

No grupo 1 estão considerados os “efeitos” do *flow*, uma espécie de diagnóstico de que o sujeito atingiu o estado de *flow*. No grupo 2 estão os “pré-requisitos” do *flow*. É neste último grupo que reside a chave do *design* de jogos.

Uma representação gráfica do *flow* nos jogos é dada pela figura 2.



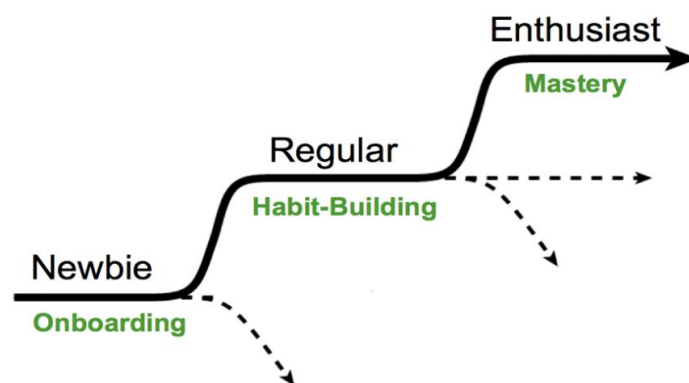
---

Figura 2: Canal do *flow* (MARCZEWSKI, 2012).

Como se pode constatar, sempre que o nível de perícia crescer mais rápido do que a dificuldade do desafio, o jogador cai no aborrecimento. O percurso contrário, em que a dificuldade excede a perícia, o jogador é conduzido à frustração. Ambos os casos provocam o abandono do jogo.

KIM (2010) defende que a experiência do jogador, tal como a dificuldade e perícia, deve variar no tempo de forma significativa. Os jogos devem proporcionar a “quantidade” certa de desafio e aprendizagem de forma a manter o jogador no limiar da sua habilidade e compenetrado na experiência. Tal como KOSTER (2013), Kim considera que os jogos são motores de aprendizagem: oferecem tarefas e competências a dominar, e recompensam-nos com desafios e oportunidades de aprendizagem ainda maiores. Pelas palavras de KOSTER (2013): *“um bom jogo é aquele que ensina tudo o que tem para oferecer, antes do jogador parar de jogar”*.

A viagem do jogador representa o percurso desde a iniciação, em que o jogador tem os primeiros contactos com o jogo, até ao nível da plena mestria (fig. 3)



---

Figura 3: A viagem do jogador (KIM, 2012)

O jogador novato começa no primeiro patamar, caracterizado pela familiarização dos conceitos e mecânicas do jogo. O segundo patamar, o do jogador regular, caracteriza-

se pela consolidação da habilidade e competências do sujeito. O terceiro e último patamar caracteriza-se pelo domínio completo do jogo. Pressupõe-se sempre um aumento no grau de dificuldade e da perícia do jogador de patamar para patamar. As indicações a tracejado representam o abandono ou a estagnação, *i.e.*, se um novato não adquirir competências suficientes acaba por abandonar o jogo, no caso de um jogador regular o abandono não é a única opção, dado que pode dar-se o caso de continuar a jogar, mesmo sem atingir o patamar da mestria.

Se esta ideia for combinada com a representação gráfica do *flow* em função da dificuldade e perícia, temos o resultado da figura 4.

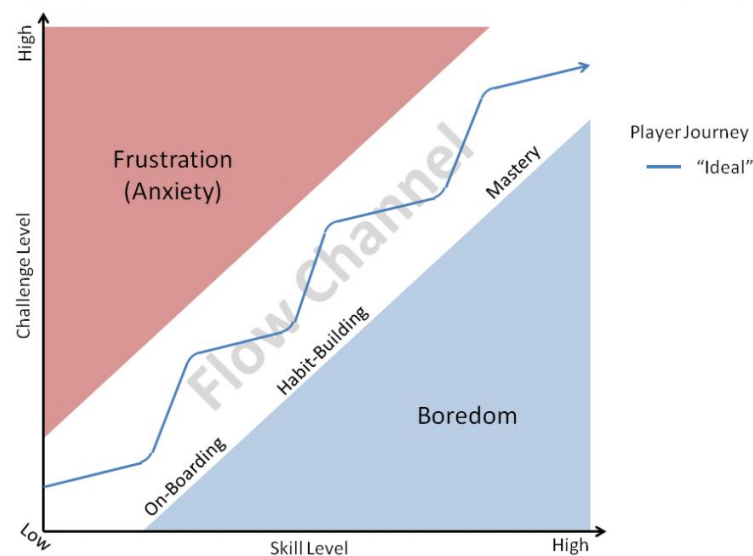


Figura 4: Flow e a viagem do jogador (MARCZEWSKI, 2012)

A linha representa a viagem ideal do jogador, sem nunca cruzar os limites da frustração ou aborrecimento. Seguir a linha da viagem do jogador significa que, em certos pontos-chave (neste caso os “máximos locais” no gráfico) há um “crescendo” mais acentuado na dificuldade do que na perícia, que culmina numa espécie de teste a toda a habilidade e aprendizagem ganhas até esse momento – por exemplo, um “Boss” num jogo de aventura.

A forma como o jogo é construído para proporcionar estes máximos varia dependendo do tipo de jogo, se bem que se seguirmos o exemplo anterior de um jogo

de aventura, a estória é um vector de criação destes momentos de tensão e portanto, do ponto de vista do *design*, um dos factores a abordar na modelação da experiência.

### 2.1.3 Os JOGOS COMO NARRATIVA

O papel da narrativa nos jogos deve ser entendido como uma forma de compreensão desencadeada e afectada pelo jogo – uma forma de o jogador aferir a sua situação no jogo. O processo principal acontece internamente no jogador (a atribuição de significado), mas pode ser direccionado por elementos narrativos formados por acontecimentos ou situações que ocorrem no jogo e que requerem algum tipo de compreensão (NITSCHKE, 2008).

Estes elementos não têm de ser necessariamente estórias, mas sim apontamentos sugestivos que encorajam o jogador a projectar significado nos eventos, personagens, objectos e espaços e criar a sua própria interpretação (SALEN e ZIMMERMAN, 2004; NITSCHKE, 2008).

O valor e a forma como a narrativa é invocada num jogo de computador depende do tipo de jogo. JUUL (2005) identifica duas componentes: progressão e emergência.

- **Progressão** – os elementos narrativos estão definidos *a priori*, como um guião que providencia as motivações para os acontecimentos e acções no jogo. Tome-se como exemplo o enquadramento do jogo *Shadow of the Colossus*<sup>1</sup>, em que o herói se aventura numa terra proibida para derrotar 16 seres gigantescos para devolver a vida a uma rapariga.
- **Emergência** – os elementos narrativos resultam da interacção do jogador com o sistema. No jogo do exemplo anterior, os elementos narrativos de emergência resultam da estratégia utilizada para defrontar cada colosso e do decorrer da batalha.

---

<sup>1</sup> *Shadow of the Colossus* é um jogo desenvolvido pela Team Ico e publicado pela Sony Computer Entertainment em 2005



A abordagem mais comum nos títulos comerciais é a combinação entre elementos de progressão e emergência: uma estrutura à base de missões em que o enquadramento narrativo está definido à partida (estória) mas que o desenrolar do jogo é feito através de elementos emergentes (JUUL, 2005; DORMANS, 2011; ANGELIDES e AGIUS, 2014).

A construção da experiência narrativa deve basear-se em 6 pontos (SALEN e ZIMMERMAN, 2004):

1. **Objectivos:** para além ajudarem o jogador a avaliar o seu progresso num jogo, ajudam o jogador a compreender o significado das suas acções no contexto narrativo;
2. **Conflito:** o conflito presume um confronto entre forças opostas. Num jogo deve sempre existir um elemento que trabalhe contra o sucesso do jogador, quer seja um vilão, outro jogador ou inerente ao sistema. Da perspectiva da narrativa, este elemento motiva e contextualiza a acção do jogador.
3. **Incerteza:** se o resultado for conhecido à partida, não há razão para jogar. A incerteza também é um conceito narrativo, porque o elemento do desconhecido injecta tensão no jogo. Incerteza no resultado é sinónimo de incerteza na narrativa.
4. **Mecânicas de jogo:** os padrões de comportamento repetitivos que compõem as mecânicas de jogo definem a experiência do jogador enquanto este passa de um elemento de estória 1 para um elemento de estória 2, *i.e.*, as mecânicas de jogo regem os segmentos emergentes, no espaço entre elementos de progressão.
5. **Espaço:** o espaço num jogo desempenha o papel fundamental de definir a forma e o enquadramento da experiência narrativa.
6. **Descritores de narrativa:** um descritor de narrativa é qualquer componente que participa no sistema de representação do jogo. Desde texto, a segmentos de vídeo, elementos da interface, objectos, música, efeitos sonoros e outros elementos audiovisuais são todos descritores de narrativa. Estes elementos devem ser cuidadosamente desenhados para maximizar a experiência do jogo enquanto narrativa.

#### 2.1.4 O PAPEL DO ESPAÇO NO DESIGN DE UM JOGO

Na secção anterior viu-se que o espaço é um dos 6 pontos a ter em conta no *design* de um jogo, enquanto elemento fundamental no enquadramento da experiência narrativa. Tome-se como exemplo o Xadrez – o tabuleiro divide o espaço de jogo em células discretas, que confinam as possibilidades de movimento das peças, ou mesmo a família de jogos *Super Mario Bros.*<sup>2</sup>, em que o espaço de jogo são percursos repletos de obstáculos, plataformas e inimigos, que separam o personagem da princesa.

SQUIRE e JENKINS (2002) defendem que as características espaciais de um jogo têm um impacto muito forte no *design* de um jogo, porque são os elementos que estabelecem os termos iniciais da experiência.

A organização do espaço determina também a influência das componentes de progressão e emergência, por exemplo em jogos de aventura, onde o jogador pode ser forçado a atravessar um corredor para atingir o próximo segmento de estória, ou ter a opção de explorar livremente um mundo aberto.

Como visto anteriormente, os objectivos de um jogo, bem como a capacidade de avaliação da evolução e progresso do jogador são factores determinantes na experiência. Este aspecto está fortemente relacionado com o facto de o espaço ser muitas vezes, directa ou indirectamente, um indicador de progresso do jogador: a posição do jogador relativamente a um objectivo diz-nos quanto é que já foi percorrido, e quanto falta para o atingir (SALEN e ZIMMERMAN, 2004; NITSCHKE, 2008; ROUSE III, 2010; NITSCHKE, 2012; ANGELIDES e AGIUS, 2014)

Assim, podemos resumir o papel do espaço no *design* de um jogo em 3 pontos:

1. Estabelece os termos iniciais da experiência;
2. Determina e condiciona a influência das componentes de progressão e emergência;
3. Funciona como um indicador do progresso do jogador.

---

<sup>2</sup> A série de jogos *Super Mario Bros.* é desenvolvida e publicada pela Nintendo desde 1985.

### 2.1.5 MAPAS EM VIDEOJOGOS

Na linguagem de videojogos o termo “mapa” é utilizado para descrever de forma geral a estrutura do espaço de jogo. Pode ser visto quer como o espaço onde decorre a acção, quer como o *layout* bi-dimensional que indica a posição do jogador relativamente a outras entidades, como obstáculos ou metas ou fronteiras (EBNER *et al.*, 2013).

Os mapas em videojogos podem assumir várias formas, sendo que uma taxonomia possível prende-se com a cardinalidade da dimensão e continuidade dos espaços (WOLF, 1997; FERNÁNDEZ-VARA *et al.*, 2005):

- quanto à cardinalidade da dimensão um mapa pode ser bidimensional (2D) ou tridimensional (3D);
- quanto à continuidade dos espaços pode ser contínuo ou discreto: um mapa é discreto se está fragmentado em zonas e existe um momento concreto de transição de uma zona para outra.

	2D	3D
Discreto	The Legend of Zelda <sup>3</sup>	Final Fantasy X <sup>4</sup>
Contínuo	(*)	The Elder Scrolls V: Skyrim <sup>5</sup>

Tabela 1: Classificação quanto ao espaço de alguns jogos de aventura/RPG.

Pode considerar-se que os RPGs (Role Playing Games) clássicos de consola (2D), como *Chrono Trigger*<sup>6</sup> ou *Final Fantasy VI*<sup>7</sup> têm várias modalidades de mapas:

- (\*)Um mapa exterior, contínuo mas desproporcionado relativamente às dimensões do personagem para poder ser atravessável;

<sup>3</sup> The Legend of Zelda é um jogo desenvolvido e publicado pela Nintendo em 1986.

<sup>4</sup> Final Fantasy X é um jogo desenvolvido pela Squaresoft e publicado em 2001 pela Squaresoft (Japão e E.U.A.) e pela Sony Computer Entertainment (Europa e Austrália).

<sup>5</sup> The Elder Scrolls V: Skyrim é um jogo desenvolvido e publicado pela Bethesda em 2011.

<sup>6</sup> Chrono Trigger é um jogo desenvolvido e publicado pela Squaresoft em 1995

<sup>7</sup> Final Fantasy VI é um jogo desenvolvido e produzido pela Squaresoft em 1994.

- Mapas das cidades/castelos/níveis, descontínuos e proporcionais relativamente às dimensões do personagem;
- Mapas de batalha, proporcionais e estáticos, onde decorre a acção baseada por turnos.

Na transição dos RPGs clássicos de consola de 2D para 3D, há um jogo que se destaca, no qual o primeiro contacto com o mapa mundo marcou muitos jogadores dessa geração:

*“Até agora, toda a acção de desenrolou na enorme cidade de Midgar. Todas as localidades – desde o reaktor de Mako, à sede da Shinra – fazem parte da mesma área. (...)*

*Neste ponto o jogo transita de 10 horas de imagens de fundo pre-renderizadas para um mapa mundo tradicional ao estilo Final Fantasy. O tipo de mapa que mostra o personagem no ecrã, com cidades e outros pontos de referência representados quase como símbolos genéricos a uma escala desproporcional.*

*Assim que esta nova forma de jogar se revela, o mini-mapa surge no canto inferior direito do ecrã. O jogador nota então que Midgar – uma avassaladora secção de 10 horas de jogo, repleta de localidades e eventos, é apenas um pequeno ponto no radar.*

*É aqui que o jogo **realmente** começa.”* (CONCELMO, 2009).



---

Figura 5: Mapa Mundo no jogo Final Fantasy VII. Fonte: CONCELMO (2009)

O jogo a que o excerto anterior se refere é o título *Final Fantasy VII*<sup>8</sup> e é o jogo mais vendido da série (WEBSTER, 2010). O jogo era distribuído via 4 CD-ROMs em grande parte devido ao volume necessário para armazenar todas as imagens de fundo e animações pre-renderizadas.

Este jogo surgiu numa altura em que o poder de processamento e armazenamento permitia obter melhorias significativas a nível do detalhe gráfico (em comparação dos os homólogos 2D), e portanto praticamente todo o conteúdo era gerado manualmente (ASHMORE e NITSCHKE, 2007).

Desde os primeiros dias dos videojogos (que recorrem a mapas) que coexistem duas formas de geração:

- Manual;
- Procedimental.

Inicialmente a geração procedimental desempenhava um papel importante, essencialmente devido ao volume de dados que os gráficos “manuais” ocupavam e

---

<sup>8</sup> Final Fantasy VII é um jogo desenvolvido pela Squaresoft e publicado em 1997 pela Squaresoft (Japão) e Sony Computer Entertainment (Europa e E.U.A.)

porque os mapas “procedimentais” garantiam experiências diferentes a cada repetição. Jogos como *Rogue*<sup>9</sup> e *Elite*<sup>10</sup> eram altamente populares no início da década de 80 precisamente por isto (ASHMORE e NITSCHKE, 2007).

Para mais, a capacidade gráfica das tecnologias disponíveis era limitada ao ponto de não oferecer diferenças significativas no nível de detalhe entre os conteúdos manuais e os procedimentais. Foi o avanço no poder de processamento e armazenamento que fez com que a produção de mapas e níveis manuais se tornasse a norma. Contudo, o aumento nos custos de produção, quer devido à dimensão das equipas de *design*, quer devido à procura de novos conteúdos, fez com que a geração procedimental voltasse cada vez mais a ser opção.

## 2.2 GERAÇÃO PROCEDIMENTAL DE CONTEÚDOS

O conteúdo dos jogos é um dos factores mais importantes para manter os jogadores imersos nos mundos virtuais. É quase garantida a capacidade de um jogo de computador apresentar ao jogador um leque de conteúdos interessantes, apenas com o senão da procura crescente por novos e interessantes conteúdos encontrar um obstáculo nos custos (em tempo e preço) da produção manual (TAKATSUKI 2007) e na sua escalabilidade (IOSUP, 2011). Em contraste com a produção manual, a Geração Procedimental de Conteúdos (PCG – do inglês *Procedural Content Generation*) é a aplicação de poder computacional para gerar, distinguir e seleccionar conteúdos com valor de entretenimento. A PCG é difícil: a maior parte dos conteúdos gerados não dependem apenas do poder computacional, mas também da capacidade de avaliar os valores técnicos e culturais dos mesmos. Surpreendentemente, após 30 anos de investigação de desenvolvimento, a comunidade ainda não construiu um gerador de aplicação generalizada (HENDRIKX *et al.*, 2013).

---

<sup>9</sup> *Rogue* é um jogo desenvolvido por Michael Toy e Glen Wichman em 1980

<sup>10</sup> *Elite* é um jogo desenvolvido por David Braben e Ian Bell e publicado pela Acornsoft em 1984

As técnicas de modelação procedimental são então uma alternativa na criação de mundos complexos, em tempo limitado e sem sobrecarregar os *designers* de conteúdo. A ideia principal por detrás da PCG é que o conteúdo não é gerado manualmente, mas sim por processos e/ou algoritmos bem definidos.

Os benefícios potenciais da PCG são:

- Geração de conteúdo de forma rápida e viável (SMITH e MATEAS 2011);
- Aumento da variedade dos conteúdos gerados (HASTINGS *et al.*, 2009; SMITH *et al.*, 2011;)
- Suporte aos jogos adaptativos centrados no jogador (LOPES e BIDARRA, 2011; YANNAKAKIS e TOGELIUS, 2011)

A amplitude destes benefícios depende no entanto do grau de controlo sobre o modelo gerador, e a capacidade de ajustar parâmetros nos modelos é desejável, para evitar perder o controlo do processo de *design* (HENDRIKX *et al.*, 2013).

Num dos principais trabalhos sobre o tema, EBERT *et al.* (2003) apontam as vantagens e desvantagem deste tipo de modelos:

- Vantagens:
  - Uma representação procedimental é relativamente compacta. O tamanho pode ser medido em *kilobytes*, enquanto que, por exemplo uma textura em ficheiro de imagem está usualmente na ordem dos *megabytes*.
  - Uma representação procedimental não tem resolução fixa. Na maioria dos casos pode apresentar-se o detalhe máximo independentemente da resolução
  - Uma representação procedimental não tem uma área fixa. Pode ter uma extensão ilimitada sem descontinuidades e sem repetição de padrões
  - Uma representação procedimental pode ser parametrizada de forma a gerar classes de conteúdos relacionados, ao invés de se limitar a um conteúdo fixo.

- Desvantagens:
  - O conteúdo procedimental pode ser difícil de construir e corrigir. A programação é geralmente difícil e programar padrões implícitos pode ser especialmente complexo;
  - O conteúdo procedimental pode ser uma surpresa. É mais seguro “prever” o comportamento de conteúdos previamente armazenados;
  - Avaliar e processar um conteúdo procedimental tem tendência a ser mais lento do que aceder a conteúdos armazenados. É um *trade-off* espaço VS tempo.
  - No caso de texturas, a suavização (*anti-aliasing*) pode levantar problemas, e é menos provável que possa ser resolvido automaticamente, comparando com imagens armazenadas.

### 2.2.1 DIMENSÕES DA GERAÇÃO PROCEDIMENTAL

Quanto ao tipo de conteúdos que pode ser gerado, TOGELIUS *et al.* (2011) apresentam uma classificação segundo 5 dimensões:

1. **Offline VS Online:** A primeira dimensão distingue se o conteúdo é gerado *online* no decorrer do jogo, ou *offline* durante a fase de desenvolvimento. Um exemplo de conteúdo *online* é quando um jogador atravessa uma porta para o interior de um edifício e o jogo gera o interior do edifício nesse instante, interior esse que até à altura era inexistente. No outro caso, existe um algoritmo que sugere *layouts* para o interiores de edifícios, que são depois escolhidos, editados e aperfeiçoados por um *designer* (humano) antes do jogo entrar em fase de produção. A geração *online* requer que o algoritmo seja muito rápido e que a qualidade dos resultados seja previsível.
2. **Conteúdo Necessário VS Opcional:** Esta dimensão está relacionada com a progressão no jogo. O conteúdo necessário é aquele que é indispensável para o jogador avançar no jogo - e.g. níveis que precisam de ser ultrapassados, adversários que precisam de ser derrotados, regras cruciais, *etc.* - enquanto o conteúdo opcional é aquele que pode ser evitado, tais como equipamentos, ou



regiões do mapa que podem ser exploradas, ou ignoradas. A principal diferença é que o conteúdo necessário tem de ser sempre correcto, *i.e.*, é inaceitável gerar um mapa de um nível que não possa ser ultrapassado, um adversário invencível, ou regras contraditórias que tornem impossível o progresso do jogador. Se o conteúdo é necessário, a dificuldade não deve estar completamente desfasada do canal do flow. Por outro lado, se o conteúdo for opcional, a validade dos conteúdos pode assumir um espectro maior e dependendo do contexto do jogo, pode variar a definição de “fracasso” do conteúdo gerado.

3. **Vectores de parâmetros VS sementes aleatórias:** Outra dimensão, relativamente ao procedimento em si, é o grau de parametrização do algoritmo. Num extremo temos o algoritmo que toma como *input* apenas os parâmetros de estado (*seed* - sementes aleatórias) para o gerador de números pseudo aleatórios. No outro extremo, o algoritmo pode receber como *input* um vector de números reais para especificar as propriedades do conteúdo a gerar. Por exemplo, um gerador de mapas pode receber como *inputs* propriedades como o número de biomas a apresentar, a rugosidade do terreno, a percentagem de cobertura de água. O contínuo sementes aleatórias - vectores de parâmetros tem o nome de “grau de controlo”.
4. **Geração estocástica VS Determinística:** Esta dimensão prende-se com a “quantidade de aleatoriedade” na geração de conteúdos. O grau de variação nos outputs de algoritmo a correr com os mesmos parâmetros é um problema de *design*. É possível (e em muitos casos desejável) que o algoritmo produza o mesmo resultado dados os mesmos parâmetros, mas a maior parte dos algoritmos não o fazem: os parâmetros de estado do gerador de números pseudo aleatórios não são considerados um parâmetro per se - isso implicaria que todos os algoritmos seriam determinísticos. Um algoritmo completamente determinístico assume a forma de compressor de dados. Por exemplo, no jogo *.kkrieger*<sup>11</sup> todas as texturas, objectos, níveis e música estão comprimidos em 96 kilobytes de espaço.

---

<sup>11</sup> *.kkrieger* é um jogo desenvolvido pela equipa *.theprodukkt* em 2004.

5. **Geração Construtiva VS Gerar-e-Testar:** A última dimensão proposta por distingue os algoritmos que podem ser chamados de construtivos e aqueles descritos como gerar-e-testar. Um algoritmo construtivo gera o conteúdo uma vez, e termina. No entanto, é necessário garantir que o conteúdo está correcto, ou pelo menos, “satisfatório” enquanto está a ser construído. Isto pode ser obtido através de procedimentos ou operações que garantidamente não geram conteúdos descontínuos, ou “partidos”. A utilização de fractais ou ruído coerente é um exemplo desta prática (MILLER, 1986.). Um algoritmo gerar-e-testar incorpora um mecanismo para gerar o conteúdo bem como um mecanismo para testar. Após a geração de um conteúdo candidato, este é submetido à avaliação de determinados critérios (*e.g.* há um percurso válido entre o ponto a e o ponto b, ou as proporções dos objectos estão no intervalo x?). Se o conteúdo não passar nos testes, é descartado total ou parcialmente e regenerado para voltar a correr os testes, e o processo termina quando forem validados todos os critérios.

### **2.2.2 CLASSIFICAÇÃO E TAXONOMIA DOS TIPOS DE CONTEÚDOS E MÉTODOS DE GERAÇÃO**

Apesar da abundância de técnicas de geração procedimental, a sua aplicação comercial não está completamente estabelecida. Os métodos têm sido aplicados com sucesso na geração de vários conteúdos, mas as soluções encontradas não são de uso generalizado.

Por um lado, a literatura disponível encontra-se dispersa por inúmeras publicações e disciplinas tais como: computação gráfica, processamento de imagem, inteligência artificial, interface homem-máquina, psicologia, linguística, geografia, ludologia entre outras. Por outro lado, HENDRIKX *et al.*, (2013) apontam que algumas das investigações e revisões não apresentam estruturas semelhantes, apesar da semelhança dos tópicos e propõem então uma estrutura desta área de investigação,

tendo em conta o espectro de tipos de conteúdos com ênfase nas técnicas que já demonstraram a sua utilidade em aplicações comerciais.

A estrutura proposta passa pela categorização e definição dos conteúdos de videojogos que podem ser gerados procedimentalmente (tabela 2) e pela apresentação de uma taxonomia dos métodos considerados fundamentais, que possam servir a geração procedimental nas várias categorias de conteúdos (tabela 3).

1. Unidades Básicas	1.1. Texturas 1.2. Sons 1.3. Vegetação	1.4. Edifícios 1.5. Comportamento 1.6. Fogo, água, pedras, nuvens
2. Espaço de jogo	2.1. Mapas interiores 2.2. Mapas exteriores 2.3. Massas de água	
3. Sistemas do Jogo	3.1. Ecossistemas 3.2. Redes viárias 3.3. Ambientes urbanos	3.4. Comportamentos de entidades
4. Cenários de Jogo	4.1. Puzzles 4.2. Storyboard 4.3. Estória	4.4. Níveis
5. Design do Jogo	5.1. Design do sistema 5.2. Design do mundo	
6. Conteúdos Derivados	6.1. Notícias e transmissões 6.2. Tabelas de classificação	

Tabela 2: Classificação dos tipos de conteúdos que podem ser gerados procedimentalmente (HENDRIKX *et al.* 2013)

1. Geradores de Números Pseudo-Aleatórios	
2. Gramáticas Geradoras	2.1. Sistemas de Lindenmayer 2.2. Gramáticas de decomposição (Split Grammars) 2.3. Gramáticas de Parede (Wall Grammars) 2.4. Gramáticas de Forma (Shape Grammars)
3. Processamento de Imagem	3.1. Morfologia Binária 3.2. Filtros de Convolução
4. Algoritmos Espaciais	4.1. Mosaicos e Camadas 4.2. Subdivisão de grelha 4.3. Vectorização 4.4. Fractais 4.5. Diagramas de Voronoi
5. Modelação e Simulação de Sistemas Complexos	5.1. Autómatos Celulares 5.2. Campos de tensores 5.3. Outros Sistemas Complexos
6. Inteligência Artificial	6.1. Algoritmos Genéticos 6.2. Redes Neurais Artificiais 6.3. Planeamento e Satisfação de Restrições

Tabela 3: Taxonomia de métodos comuns de geração de conteúdos para jogos (HENDRIKX *et al.*, 2013)

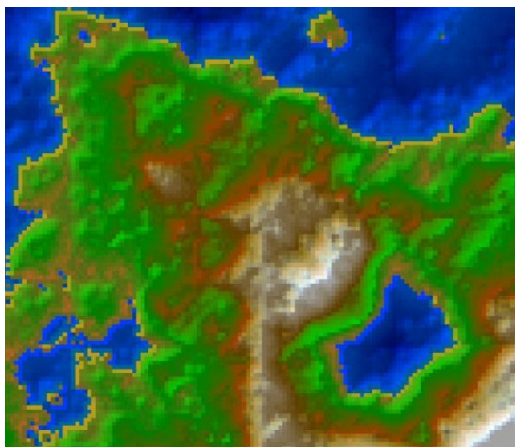
Para enquadrar as próximas secções com os objectivos do trabalho, será dado particular ênfase a determinado tipo de conteúdo e métodos.

O tipo de conteúdo que irá ser focado será o Espaço de Jogo, mais concretamente Mapas Exteriores e Massas de Água. No que diz respeito aos métodos, serão abordados Geradores de Números Pseudo-Aleatórios através da exploração da produção de ruído coerente e Algoritmos Espaciais, mais concretamente Fractais.

### 2.2.3 ESPAÇO DE JOGO:

A geração de mapas exteriores tem como base a produção de mapas de altimetria (SMELIK *et al.*, 2009). Os algoritmos mais antigos são baseados em métodos de subdivisão: uma matriz grosseira de altimetria é iterativamente subdividida para

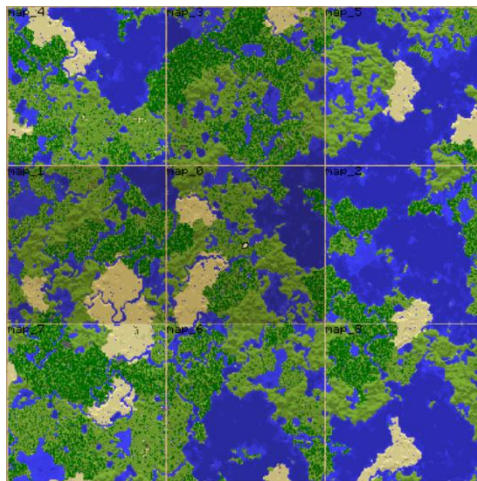
acrescentar detalhe. Dentro deste tema, MILLER (1986) lista e apresenta algumas variantes do algoritmo de deslocamento do ponto central (fractal).



---

Figura 6: Mapa gerado através do algoritmo de deslocamento do ponto central (Fonte: <http://www.avanderw.co.za/>)

Actualmente, os métodos mais populares envolvem funções de ruído. Num exemplo de sucesso recente, o título *Minecraft*<sup>12</sup>, todo o mundo é gerado à base de ruído Perlin (PERSSON, 2011)



---

Figura 7: Exemplo de um mapa do jogo Minecraft (fonte: [www.minecraftforum.net](http://www.minecraftforum.net))

---

<sup>12</sup> Minecraft é um jogo desenvolvido por Markus Persson em 2011.

A geração de massas de água tais como rios, lagos e oceanos ocorre durante, ou imediatamente após o produção do mapa de altimetria (KELLEY *et al.*, 1988; PRUSINKIEWICZ e HAMMEL, 1993; BELHADJ e AUDIBERT, 2005). O método mais simples utilizado é arbitrar uma cota no mapa, abaixo da qual se consideram áreas submersas.

Uma vez gerada a altimetria e massas de água, funções adicionais podem gerar superfícies de temperatura, humidade, tipos de solo e precipitação utilizadas para a definição de biomas, maior parte das vezes através da modificação/adaptação do diagrama de Whittaker (figura 8) (PATEL, 2010; PERSSON, 2011; ANGELIDES e AGIUS 2014).

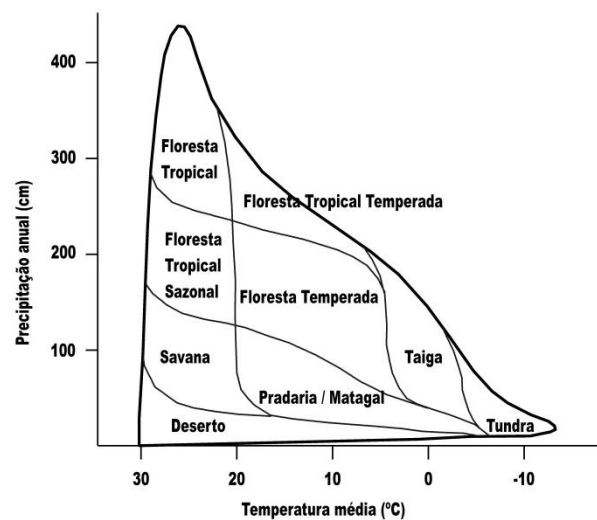


Figura 8: Biomas. Adaptado de WHITTAKER (1975).

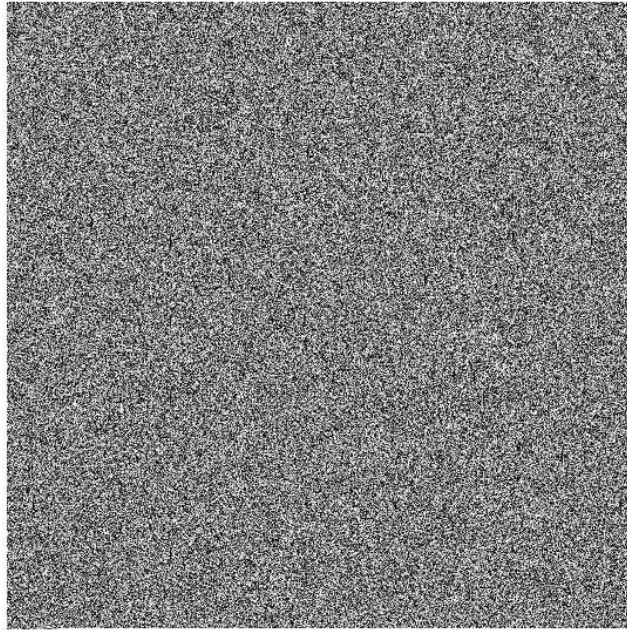
#### 2.2.4 GERADORES DE NÚMEROS PSEUDO ALEATÓRIOS

*“A geração de números aleatórios é demasiado importante para ser deixada ao acaso”*  
(COVEYOU, 1969)

Geradores de números pseudo-aleatórios são algoritmos que geram sequências de números que se aproximam das propriedades da aleatoriedade.

Os números gerados são o resultado de um cálculo determinístico baseado no estado prévio do gerador. Se o gerador for inicializado com os mesmos parâmetros de estado (“seed”) a sequência de números é sempre igual.

Uma superfície gerada desta forma tem o seguinte aspecto:



---

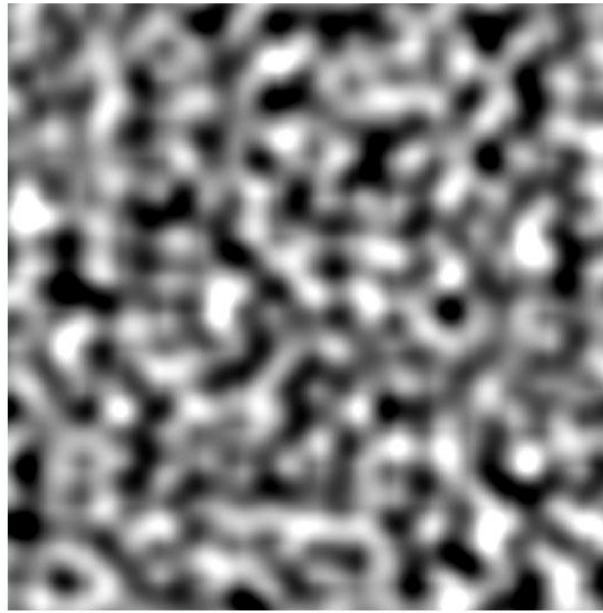
Figura 9: Ruído Branco.

A figura 9 representa o chamado “ruído branco”: números aleatórios, uniformemente distribuídos sem qualquer correlação entre números sucessivos.

Seja  $x$  a posição e  $f(\cdot)$  a função de ruído: não existe correlação entre  $f(x_1)$  e  $f(x_2)$  a não ser que  $x_1 = x_2$ .

Sob esta forma, o ruído tem uma utilidade muito reduzida na geração procedimental.

Os métodos empregues na geração procedimental apresentam um aspecto mais “suavizado”, ou espacialmente correlacionado:  $f(x_1) \approx f(x_2)$  se  $x_1$  for um ponto próximo de  $x_2$ , e tomam a designação de ruído coerente (figura 10):



---

Figura 10: Ruído Coerente (ruído Perlin)

As propriedades desejadas do ruído coerente são as seguintes:

- Tem de ser uma função pseudo-aleatória repetível: para valores de *input* iguais, deve devolver o mesmo *output*;
- Tem domínio definido, por exemplo  $[0, 1]$ ;
- Não tem periodicidade aparente ou padrões regulares. As funções pseudo-aleatórias são sempre periódicas, mas o período pode ser definido como muito grande;
- Tem de ser estacionário: invariante face à translação;
- Tem de ser isotrópico: invariante face à rotação.

#### 2.2.4.1 RUÍDO PERLIN

Para os efeitos deste trabalho, a função de ruído coerente a considerar será a desenvolvida por PERLIN (1985) fruto do trabalho do “*Mathematical Applications Group, Inc*” para o filme *Tron*, da *Disney*. O Ruído Perlin, como ficou conhecido,



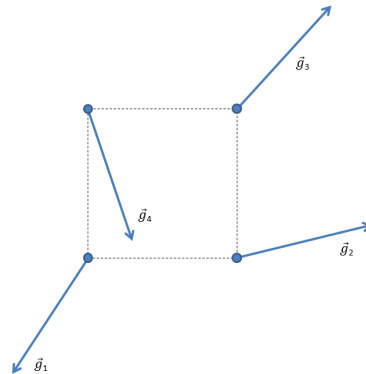
consiste numa função que calcula o valor de ruído na posição  $x$  com base numa grelha em que os vértices se encontram nas coordenadas inteiras, e a cada vértice está associado um vector gradiente aleatório. O ruído na posição  $x$  resulta da interpolação bicúbica dos produtos internos entre os vectores gradiente e a posição de  $x$  (EBERT *et al.*, 2003).

Para uma melhor compreensão, a explicação acima vai ser decomposta em 3 fases:

- Grelha de gradientes;
- Cálculo dos produtos internos;
- Interpolação.

#### Grelha de gradientes:

A grelha de gradientes consiste na atribuição de um vector aleatório a cada coordenada inteira. Estes vectores são normalizados, *i.e.*, a norma é 1, apenas variando a direcção, sentido e ponto de aplicação (figura 11).

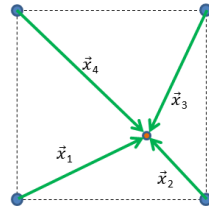


---

Figura 11: Pormenor de uma quadrícula da grelha de gradientes.

#### Cálculo dos produtos internos:

Para cada posição  $x$ , são identificados os 4 vértices da respectiva quadrícula, e encontrados os vectores de posição  $x$  relativamente a cada um desses vértices (figura 12).




---

Figura 12: Vectores da posição de  $x$  relativamente aos vértices da grelha.

Posteriormente são calculados os produtos internos dos vectores de gradiente e posição com o mesmo vértice de origem:

$$v_i = \vec{g}_i \cdot \vec{x}_i \quad (1)$$

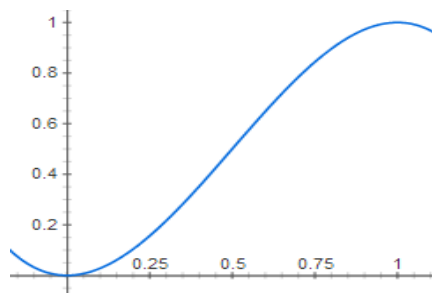
Por definição, nas coordenadas inteiras o valor do produto interno é 0.

#### Interpolação:

Calculados os 4 produtos internos correspondentes a cada vértice, é interpolado o valor de ruído na posição  $x$  através de uma função da forma:

$$f(x) = 3x^2 - 2x^3 \quad (2)$$

Que produz uma variação suavizada, como ilustrado na figura 13 abaixo:




---

Figura 13: Aspecto da função de interpolação.

A duas dimensões, e considerando os valores dos 4 vértices, a função de interpolação é aplicada 2 vezes no eixo horizontal ( $x$ ) e uma vez no eixo vertical ( $y$ ).

### 2.2.5 FRACTAIS

Muitos, se não a maioria dos padrões da Natureza são irregulares e fragmentados. Do ponto de vista geométrico, a Natureza não exhibe apenas um grau mais elevado, como está num nível totalmente diferente de complexidade. O número de escalas distintas e extensão dos padrões naturais é para todos os efeitos práticos, infinito. MANDELBROT (1979) concebeu e implementou uma nova geometria da natureza, que descreve muitos desses padrões irregulares e fragmentados, identificando uma família de formas a que chamou de fractais (do latim *fractus*: fragmento + álgebra: do árabe *al-jabr* para reunião de partes).

MUSGRAVE (1989) descreve fractais como objectos geometricamente complexos, complexidade essa que surge da repetição da mesma forma em todas as escalas.

Enquanto a geometria euclidiana tem dimensões inteiras (0 – ponto; 1 – linha; 2 – plano; 3 - espaço), os fractais têm a peculiar propriedade da dimensão fractal, que pode assumir valores reais tais como 2,3. A parte inteira da dimensão fractal, o “2” no 2,3 indica a dimensão euclidiana subjacente – o plano neste exemplo. A parte fraccionária, o “,3” é chamado de incremento fractal. À medida que o incremento fractal varia entre “,0” e “,9999(9)”, o fractal passa de plano, a ocupar parcialmente a dimensão seguinte – o espaço neste exemplo – ficando mais enrugado quanto maior o incremento fractal (MANDELBROT, 1983; MUSGRAVE, 1989).

#### 2.2.5.1 MOVIMENTO FRACCIONAL BROWNIANO

O movimento fraccional browniano (fBm do inglês *fraccional brownian motion*) providencia uma forma de gerar superfícies fractais (MANDELBROT, 1983; FOURNIER *et al.*, 1982; MUSGRAVE, 1991).

Do ponto de vista da aplicação, o fBm consiste em calcular a soma ponderada de, por exemplo, funções de ruído no ponto  $x$  (equação 3):

$$fBm(x) = \sum_{i=0}^{n-1} N(x \cdot L^i) L^{-iH} \quad (3)$$

Em que:

- $N(x)$  = função de ruído na posição  $x$ ;
- $L$  = Lacunaridade: é a variação que a frequência sofre em cada iteração. Na prática, a lacunaridade assume sempre o valor 2.0 – a frequência duplica;
- $n$  = nº de oitavas: em acústica, duplicar a frequência (que é a implicação de ter lacunaridade 2.0) aumenta precisamente uma oitava. Daí a nomenclatura de oitavas na formação de fractais por esta via - correspondem ao número de vezes que se escalou e somou a função de ruído para adicionar detalhe.
- $H = 1$  – incremento fractal: como já se referiu anteriormente, o incremento fractal influencia a rugosidade da função. Um aumento do incremento fractal aumenta a rugosidade. Da forma que  $H$  está definido, a variação da rugosidade é a oposta – aumenta à medida que  $H$  varia de 1.0 para 0.0 aproximando-se de ruído branco no limite (figura 14)

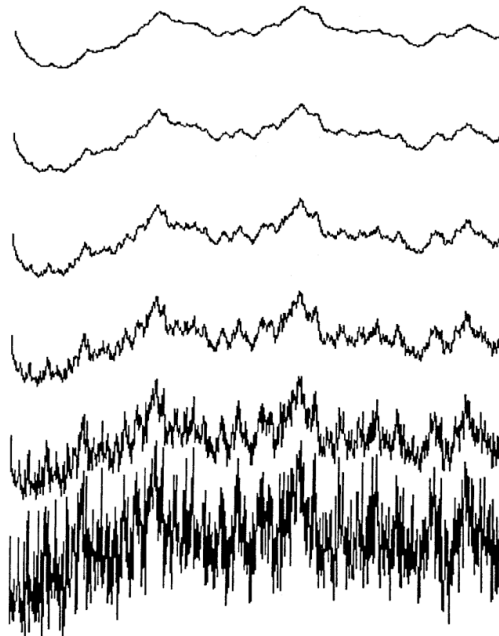


Figura 14: fBm a uma dimensão, com  $H$  a variar de 1.0 (mais suave) no topo, para 0.0 (mais rugoso) em intervalos de 0.2. (EBERT *et al.*, 2003)

Do ponto de vista prático,  $H$  determina a variação da amplitude de oitava para oitava, ou seja, o ganho.

$$ganho = L^{-iH} \quad (4)$$

Por exemplo, se  $H = 1$  e assumindo a lacunaridade típica 2.0 vamos ter ganho  $2^{-i}$ , que implica que a cada oitava, a amplitude reduz-se para metade.

A figura 15 exemplifica a duas dimensões a aplicação de fBm.

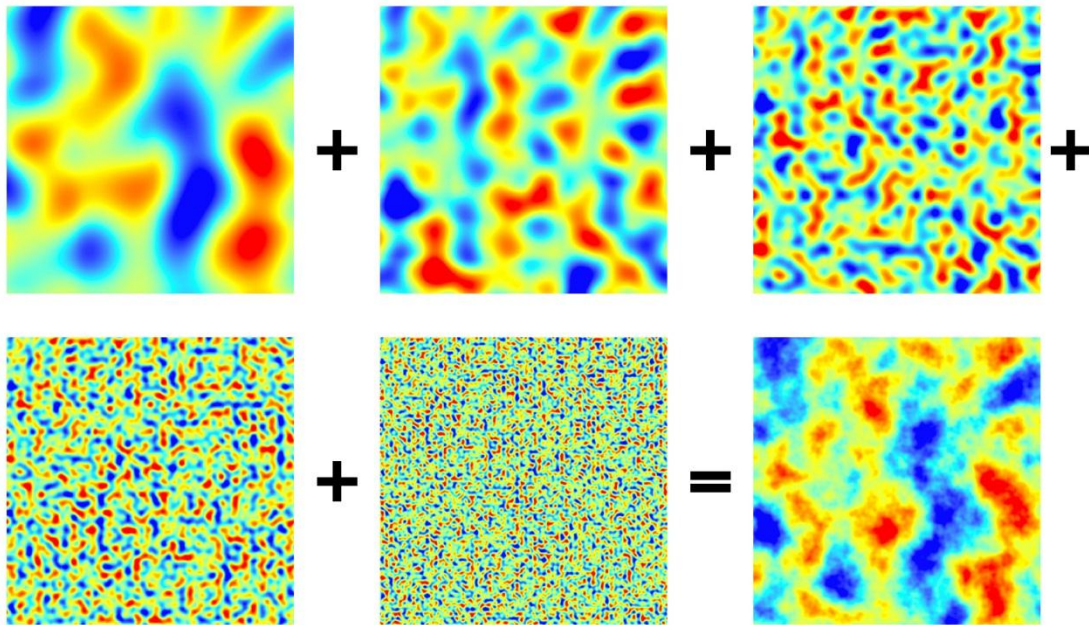


Figura 15: fBm a duas dimensões.  $H = 1.0$  (ganho 0.5), Lacunaridade = 2.0 e oitavas = 5.

#### 2.2.5.2 DESLOCAMENTO DO PONTO CENTRAL

O algoritmo de deslocamento do ponto central, ou a variação aqui apresentada, diamante-quadrado, é um método de subdivisão bastante rápido e de fácil compreensão e implementação (FOURNIER et al 1982; MILLER, 1986).

Cada iteração da subdivisão é composta por 2 passos (figura 16):

- Passo diamante: considera os 4 vértices de um quadrado, o valor do ponto central é calculado através da média dos valores dos 4 vértices, mais uma quantidade aleatória. Este passo gera os “diamantes” quando temos vários quadrados numa matriz.
- Passo quadrado: considera os 4 vértices de um diamante e calcula o valor do ponto central através da média dos valores dos 4 vértices e soma um valor aleatório, tal como no passo diamante. Este passo gera novos quadrados, que vão dar origem ao passo diamante da iteração seguinte.

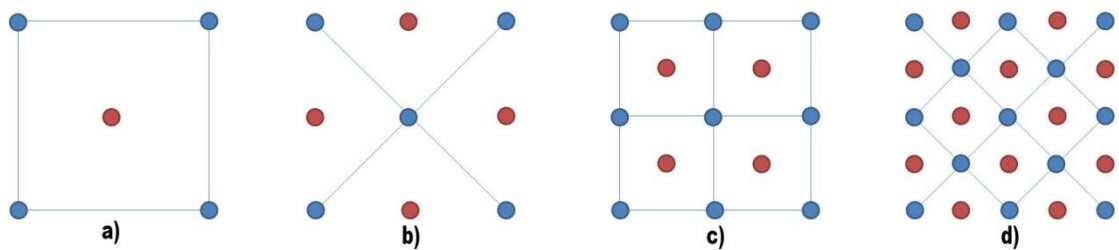


Figura 16: Esquema do algoritmo diamante-quadrado. a) determinação do primeiro ponto central; b) 1º passo diamante; c) 1º passo quadrado; d) 2º passo diamante

O resultado é uma superfície do género da figura 17 abaixo:

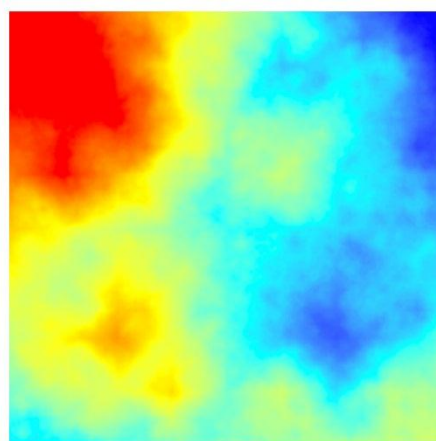


Figura 17: Superfície gerada através do algoritmo diamante-quadrado.

## 2.3 SISTEMAS DE INFORMAÇÃO GEOGRÁFICA

O conceito de Sistema de Informação Geográfica (SIG) é bastante abrangente, no sentido em que alberga um grande conjunto de tecnologias, processos e metodologias com aplicações nas mais variadas áreas. O espectro de aplicações vai desde áreas mais óbvias como a engenharia, planeamento e gestão, transportes e logística, a áreas como os seguros, desenvolvimento de negócio e marketing (MALIENE *et al.*, 2011).

BURROUGH (1986) definiu um SIG como “um conjunto de ferramentas para recolher, armazenar, transformar e visualizar dados de natureza espacial do mundo real, para um conjunto particular de objectivos”.

Esta definição pode ser alargada tal como GOODCHILD e KEMP (1990) propuseram, para os quais um SIG é constituído por um sistema de “*hardware, software e procedimentos desenhado para realizar a captura, armazenamento, manipulação, análise, modelação e apresentação de dados referenciados espacialmente para a resolução de problemas complexos de planeamento e gestão*”.

Um ponto comum entre as definições de SIG, é a sua orientação para diferentes objectivos e tipos de utilização. É neste sentido que MAGUIRE (1991) distingue 3 perspectivas diferentes:

- Cartográfica: perspectiva de um SIG como sistema de processamento de visualização de mapas (TOMLIN, 2013);
- Base de dados: SIG como ferramenta de gestão de bases de dados geográficas (FRANK, 1988);
- Análise Espacial: SIG como ferramenta analítica para a produção de informação/conhecimento (GOODCHILD, 1988; OPENSHAW, 1991; GOODCHILD, 1992).

### **2.3.1 MODELOS DE DADOS**

Quanto às características da informação geográfica, esta tem duas componentes fundamentais (GOODCHILD, 1992; FALCONER e FORESMAN, 2002):

- Componente espacial: relacionada com a forma e posição dos elementos geográficos e relações espaciais entre estes;
- Componente descritiva: propriedades e atributos alfanuméricos associados aos elementos geográficos.

Segundo GOODCHILD (1992), da perspectiva de um SIG, a informação geográfica pode ser conceptualizada de duas formas: ou o mundo é representado com um conjunto de camadas, em que cada uma define a variação contínua ou quasi-contínua de uma variável; ou o mundo é representado como um espaço vazio, populado por objectos discretos.

Estas duas formas correspondem respectivamente aos modelos Raster / Matricial e Vectorial (figura 18):

O modelo Raster consiste numa matriz de pixéis ou células de uma determinada dimensão organizados em linhas e colunas, em que a cada célula corresponde um valor que representa informação ou variável, tal como temperatura ou altimetria. A dimensão da célula determina a resolução do modelo: quanto menor a dimensão da célula, maior a resolução.

O modelo Vectorial, mais adequado para representar elementos discretos, baseia-se em 3 tipos de entidades:

1. Pontos: definidos por pares de coordenadas;
2. Linhas: representam elementos lineares através de segmentos de recta ou arcos;
3. Polígonos: representam áreas ou regiões, e são delimitados por um conjunto de linhas.

O ponto é a primitiva do modelo, já que as linhas correspondem à ligação entre pontos, e os polígonos correspondem à área definida por um conjunto de linhas.



A cada objecto dentro de cada uma destas entidades podem ser adicionadas propriedades ou atributos correspondentes a vários temas.

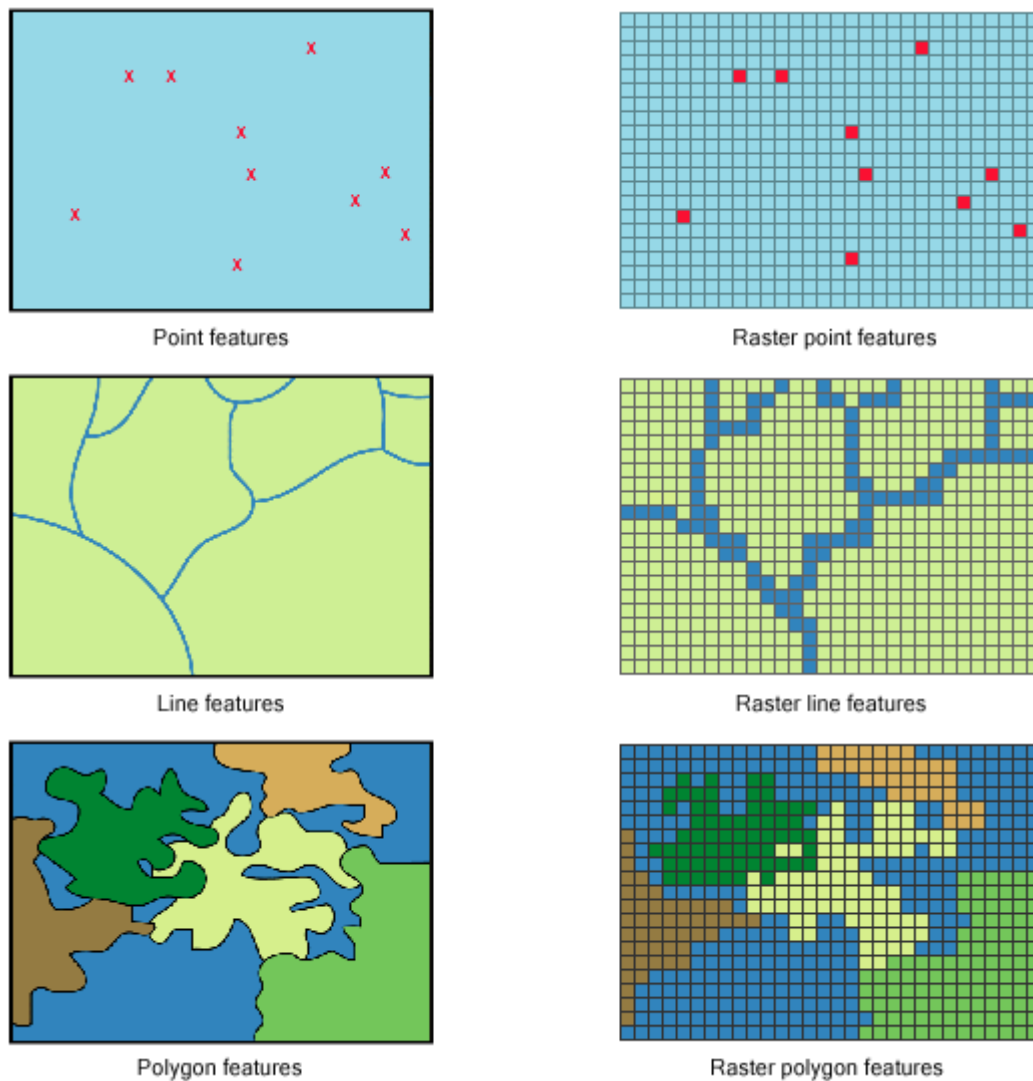


Figura 18: Representação de entidades pelo modelo Vectorial (à esquerda) e Raster (à direita) (ESRI, 2012).

Ambos os modelos apresentam vantagens e desvantagens, sendo que no final, o modelo Vectorial apresenta melhores características para representar objectos discretos com níveis de precisão elevados, enquanto que o modelo Raster é mais indicado em processos de modelação geográfica e análise espacial.

### 2.3.2 MODELOS DIGITAIS DO TERRENO

O conceito de Modelo Digital do Terreno (MDT) foi introduzido por MILLER e LAFLAMME (1958) como “simplesmente uma representação estatística da superfície contínua do terreno, através de um grande número de pontos de coordenadas ( $x, y, z$ ) conhecidas, num espaço de coordenadas arbitrário”.

WEIBEL e HELLER (1993) simplificam esta definição, referindo que um MDT é a representação digital de uma parte da superfície terrestre.

Um MDT pode ser a representação da distribuição espacial de uma ou mais variáveis que caracterizam o terreno. Uma vez que é composto pela localização bidimensional, mais a representação matemática da(s) variável(s), é normalmente visto como um entidade a 2,5 dimensões no espaço geográfico tridimensional (LI *et al.*, 2010).

As operações de modelação e processamento de um MDT (generalizadas na figura 19) devem ser vistas como componentes de um SIG funcionalmente distintas, mas relacionadas com as restantes operações a duas dimensões.

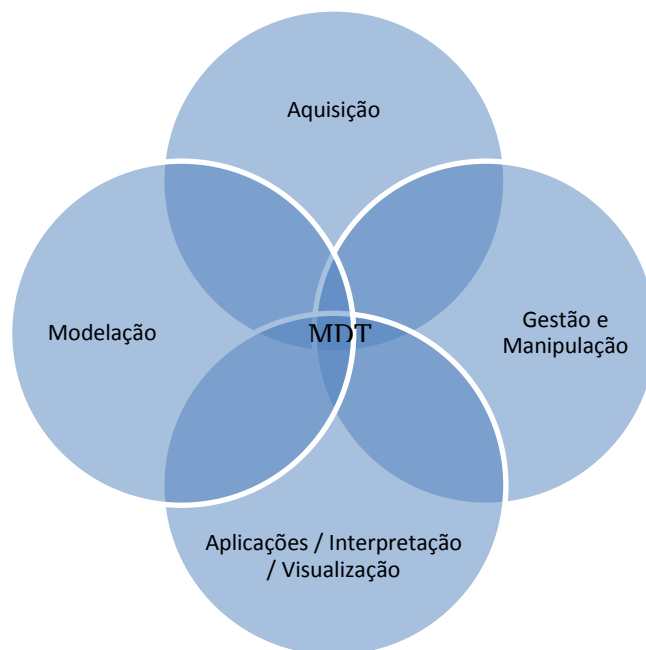


Figura 19: Componentes da Modelação Digital do Terreno. Adaptado de LI et al (2010)

O diagrama da figura 19 não apresenta uma estrutura linear pelas relações e o envolvimento das várias disciplinas em cada um dos componentes:

- **Aquisição:** *e.g.* fotogrametria, topografia, detecção remota e cartografia;
- **Modelação:** *e.g.* fotogrametria, topografia, cartografia, matemática, geografia, geometria, processamento de imagem;
- **Gestão e Manipulação:** *e.g.* bases de dados, codificação e compressão, algoritmos e estruturas de dados, computação gráfica;
- **Aplicações:** *e.g.* engenharias (civil, geológica, minas, ambiental,...), fotogrametria, topografia, detecção remota, geografia, geomorfologia, arquitectura paisagista.

Uma disciplina como fotogrametria está directamente envolvida na aquisição de dados, mas por sua vez utiliza MDTs para orto rectificação de fotografias aéreas ou imagens de satélite.

Algumas aplicações generalizadas que utilizam MDTs passam por (Li et al, 2010):

- Ordenamento do território;
- Análise e simulações hidrográficas;
- Análise da visibilidade;
- Análise de declives, exposição e cálculo de volumes;
- Análises geomorfológicas e erodibilidade do solo;
- Interpretação e processamento de imagens de detecção remota;
- Treino e Simulação Militar (MS&T do inglês Military Simulation and Training);
- Videojogos.

Um MDT pode apresentar-se de acordo com os modelos de dados de um SIG, sendo que o que as distingue é (para além do modelo conceptual) a forma como são obtidas, condicionada pela própria morfologia do terreno e o fim a que se destina (figura 20):

- Modelo matricial (ou grid) – este modelo apresenta uma estrutura matricial de altimetrias que reflecte a forma mais simples de armazenamento, do tipo (x,y,z).
- Modelo vectorial – resulta da criação de uma rede irregular de triângulos (TIN – do inglês Triangulated Irregular Network) cujos vértices correspondem aos pontos de amostragem.

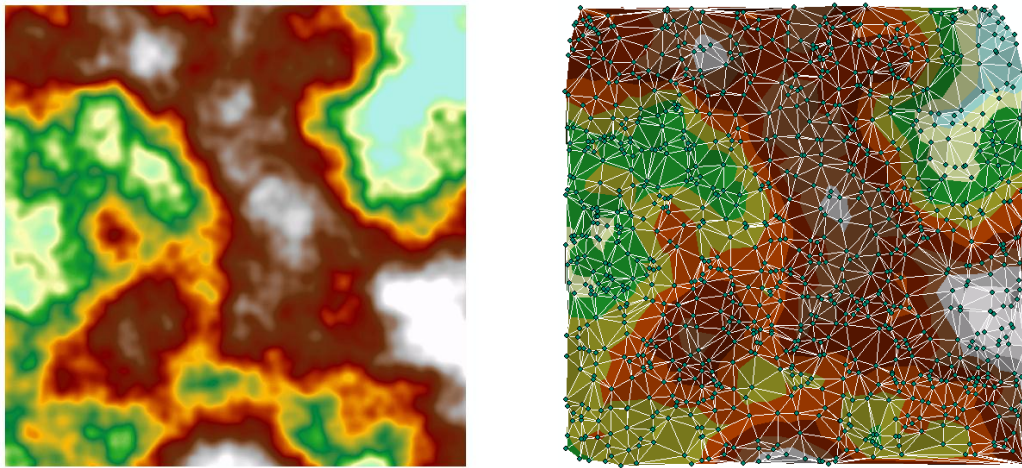


Figura 20: Modelo digital do terreno: Grid à esquerda; TIN à direita.

No modelo matricial, as relações topológicas entre os registos estão definidas implicitamente, e dada a forma de armazenamento, a manipulação deste modelo é simples e os algoritmos utilizados têm tendência a ser bastante directos. Por outro lado, a densidade dos pontos de amostragem numa grelha regular pode não ser adequada para a representação precisa de alguns elementos do terreno (Li *et al.*, 2010)

Nos modelos vectoriais, dado que a sua composição se baseia em elementos triangulares, os vértices reflectem a densidade variável dos pontos de amostragem, o que permite aferir acerca da maior ou menor precisão em determinadas regiões. Ao contrário do modelo matricial, as relações topológicas não são implícitas e devem ser calculadas e registadas explicitamente, tornando os modelos vectoriais mais complexos e comprometendo a eficiência de alguns algoritmos matriciais (WEIBEL e HELLER, 1993).

Idealmente, um MDT deve ser suficientemente flexível para servir vários propósitos, independentemente do modelo de dados utilizado.

### **2.3.3 ANÁLISE ESPACIAL**

A análise espacial é um conjunto de métodos, cujos resultados variam, consoante a posição dos objectos em análise varia (LONGLEY *et al.*, 2005).

Corresponde no fundo a um conjunto de ferramentas analíticas que entram em conta com o enquadramento espacial, reproduzindo mais fielmente as relações implícitas da informação geográfica, que a investigação mostrou não ser possível de alcançar através de metodologias não espaciais (MILLER, 1999).

Além do mais, praticamente toda a informação tem uma componente espacial mais ou menos precisa, e o modelo de análise não espacial é apenas um caso específico do modelo espacial, o que torna o modelo de análise espacial o mais geral (CRESSIE, 1993)

Para GOODCHILD (1988), é nas capacidades de análise espacial que o verdadeiro valor de um SIG reside. É através da análise espacial que um SIG providencia produtos de valor acrescentado, a partir da informação existente.

Quanto às operações de análise espacial, várias estruturas de classificação das mesmas têm sido oferecidas, umas de pontos de vista mais conceptuais, outras mais pragmáticas ou mais relacionadas com determinado modelo de dados (CRESSIE, 1993; BERRY, 2004; LONGLEY *et al.*, 2005; TOMLIN, 2013).

A classificação apresentada abaixo (BERRY, 2004), foi escolhida pela sua simplicidade, flexibilidade e adaptabilidade quer a modelos mais físicos, quer a modelos mais abstractos, e também porque toca nas operações fundamentais sobre modelos Raster que se pretendem abordar metodologicamente.

Sendo assim, as operações a considerar estão divididas em quatro classes:

- **Reclassificação:** atribuição de novos valores a uma superfície existente, com base no seu valor inicial;
- **Distâncias:** criação de superfícies que expressam a distância entre localizações;
- **Sobreposição:** criação de novas superfícies, onde o valor em cada posição é calculado como função dos valores independentes associados a essa posição num ou mais mapas;
- **Vizinhança:** criação de novas superfícies cujos valores em determinada posição são calculados com base nos valores das posições “vizinhas”.

No âmbito da metodologia a ser empregue, há um conjunto de operações de vizinhança que requer especial atenção neste capítulo, e dizem respeito á modelação hidrológica.

Os algoritmos utilizados na modelação hidrológica (presentes no *software* que irá ser utilizado) são baseados nos trabalhos de JENSON e DOMINGUE (1988) e TARBOTON *et al.* (1991).

Para assegurar o desempenho correcto dos algoritmos e continuidade na rede de drenagem, é necessário eliminar áreas de drenagem interna (*sinks*) e picos que possam estar presentes no mapa de altimetria (figura 21):

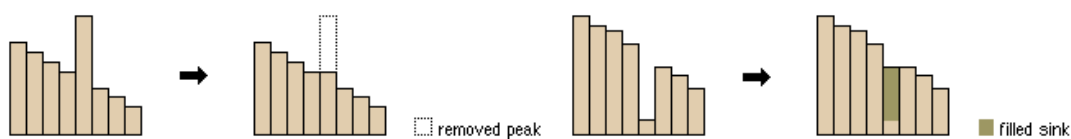


Figura 21: Fill - Remoção de picos e áreas de drenagem interna (ESRI, 2012)

Após removidas as fontes de descontinuidade, é determinada a direcção do escoamento através da identificação do maior declive descendente de cada célula, relativamente às 8 células vizinhas.

O maior declive descendente (*mdd*) é determinado através da fórmula:

$$mdd = \frac{\Delta z}{distância} * 100 \quad (5)$$

Em que  $\Delta z$  é a variação de cota, e a distância é calculada entre os centroides de cada célula, *i.e.*, se a célula tiver lado 1, a distância entre células ortogonais é 1, e a distância entre duas células diagonais é  $\sqrt{2}$ . Quando é encontrada o maior declive descendente, a célula é codificada com o valor que corresponde a essa direcção (figura 22):

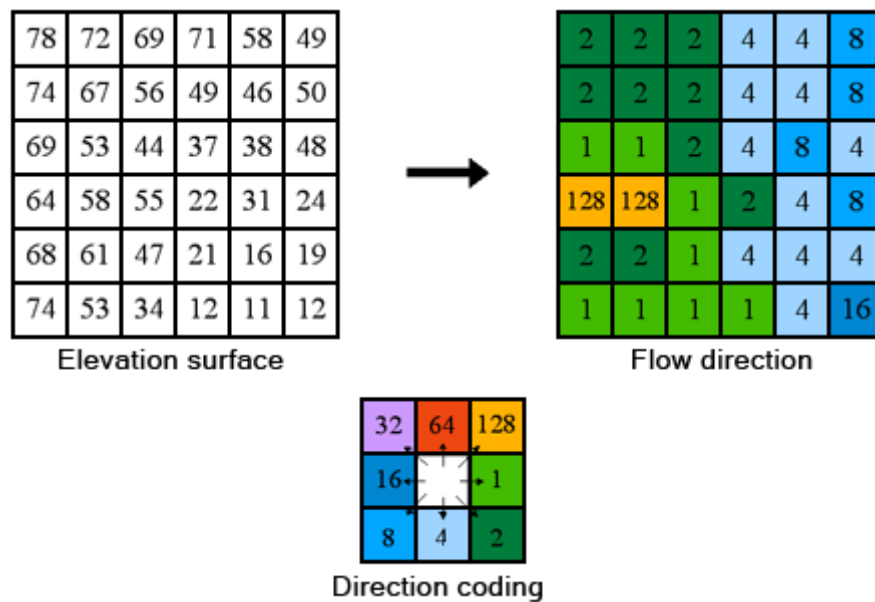


Figura 22: Direcção do escoamento (ESRI, 2012)

Após determinada a direcção do escoamento para cada célula, determina-se a acumulação do escoamento em cada uma delas. Para cada célula, é contado e acumulado o número total de células que escoam naquela direcção (figura 23).

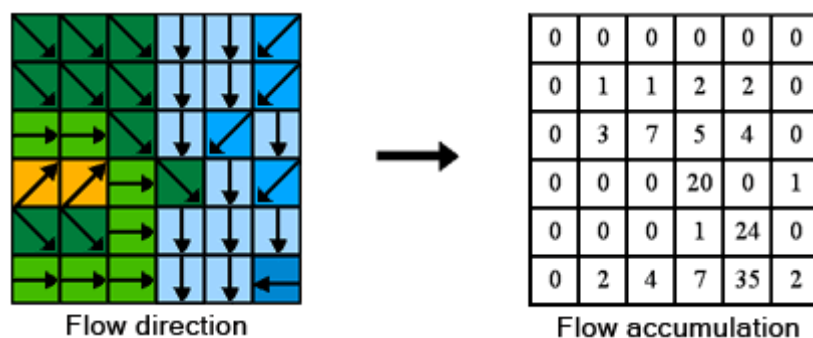


Figura 23: Acumulação do Escoamento (ESRI, 2012).

## 2.4 SISTEMAS DE INFORMAÇÃO GEOGRÁFICA E VIDEOJOGOS

As potencialidades “naturais” de um SIG para modelar o mundo, e a importância e representatividade do espaço enquanto conteúdo de um videogame indiciam uma convergência entre as indústrias. No entanto, antes de elaborar acerca do potencial para a transferência de tecnologias dos videogames para o *software* de SIG e vice-versa, é importante reconhecer que há diferenças significantes entre ambas.

PAGULAYAN *et al.* (2003) identificam 10 características que distinguem os videogames e as aplicações de produtividade (onde se inserem as aplicações de SIG):

Característica	Videojogos	Aplicações de SIG
<b>Propósito</b>	Entretenimento	Produtividade
<b>Objectivos</b>	Intrínsecos	Definidos pelo utilizador
<b>Software</b>	Muitas alternativas	Poucas alternativas
<b>Consistência</b>	Baixa (variabilidade)	Alta (familiaridade)
<b>Restrições</b>	Impostas	Evitadas
<b>Sinais não visuais</b>	Definem o ambiente	Funcionais (funções/alertas)
<b>Pontos de vista</b>	Integrados	Raros
<b>Consumidores</b>	Indivíduos	Organizações
<b>Inovação</b>	Desejada	Cautelosa
<b>Hardware</b>	Variado/Específico	Padronizado/Multiplataforma

Tabela 4: Características diferenciadoras de SIG e Videogames. Adaptado de PAGULAYAN (2003)



A tabela 4 mostra que a tecnologia associada aos videjogos está mais vocacionada para a variabilidade e criatividade, enquanto que as aplicações SIG têm características mais conservadoras.

Apesar destas diferenças ao nível das aplicações, mapas e jogos têm uma longa história de co-evolução, e depois de muitos anos de desenvolvimentos paralelos e por vezes independentes, é possível apontar 5 grandes áreas de convergência entre as tecnologias (AHLQVIST, 2011):

1. Análise Espacial;
2. Simulação;
3. Aumento do Realismo e Mundos Virtuais;
4. Construção de Mundos;
5. Conteúdos Padronizados.

As áreas da Simulação, Aumento do Realismo e Mundos Virtuais e Conteúdos Padronizados podem ser considerados pontos de convergência mais tecnológicos, enquanto a Análise Espacial e Construção de Mundos são os dois temas onde a intersecção é mais evidente, e sobre os quais se debruça a tese.

Os trabalhos mais comuns em que se verifica a interacção entre SIG e videjogos são no domínio do Aumento do Realismo e Mundos Virtuais, através de aplicações de visualização de informação geográfica com recurso a motores de jogo (FRITSCH e KADA, 2004; FRIESE *et al.*, 2008; SHEPHERD e BLEASDALE-SHEPHERD, 2009; MAT *et al.*, 2014). Estes trabalhos procuram as potencialidades de representação gráfica dos motores de jogo para alargar a compreensão e leitura dos mapas para além das duas dimensões. É no entanto nestes estudos que se verifica que no ponto de convergência dos Conteúdos Padronizados ainda há muito a percorrer: No levantamento de FRITSCH e KADA (2004) é apontado o obstáculo do formato de dados, e passados 10 anos, MAT *et al.* (2014) encontram o mesmo problema.

No domínio da Análise Espacial, as metodologias comuns a ambas as tecnologias passam, por exemplo, pela modelação digital do terreno, indexação espacial,

operações de buffering (usada na detecção de colisões), análise de redes para a determinação de percursos (SHEPHERD e BLEASDALE-SHEPHERD, 2009).

Mais voltados para uma perspectiva analítica, KANG e KIM (2014) referem uma abordagem SIG para monitorizar ao longo do tempo os padrões de acção dos jogadores e aumentar a eficiência da produção de conteúdos.

Numa perspectiva de Construção de Mundos, SILVA e COELHO (2010) e CULLEN e O'SULLIVAN (2011) exploram ferramentas de modelação (gramáticas geradoras) integradas com bases de dados geográficas que permitem a geração de ambientes urbanos com base em dados existentes.

Num misto de Análise Espacial com Construção de Mundos, PARBERRY (2013; 2014) apresenta uma abordagem que modifica ligeiramente o algoritmo de ruído Perlin, de forma a considerar as funções de distribuição da altimetria de localizações geográficas reais armazenadas na geração de terreno aleatório com características semelhantes a essas localizações.

A metodologia proposta consiste na utilização de uma aplicação desktop de SIG para gerar espaços de jogo de videojogos, com base nos procedimentos mais comuns. Segundo as dimensões de TOGELIUS *et al.* (2011), trata-se de uma geração:

- offline;
- de conteúdo necessário;
- segundo sementes aleatórias;
- estocástica;
- construtiva.

Segundo a classificação de HENDRIKX *et al.* (2013) o tipo de conteúdo corresponde aos Mapas Exteriores e Massas de Água, e os métodos enquadram-se nas categorias de Geradores de Números Pseudo Aleatórios e Algoritmos Espaciais.

No que diz respeito a algoritmos espaciais, as aplicações Desktop de SIG já estão munidas de um conjunto de ferramentas que possibilitam as operações necessárias de análise espacial. É na capacidade de gerar superfícies pseudo-aleatórias que reside a lacuna destas aplicações, mas que pode ser colmatada com recurso às linguagens de programação já integradas nas mesmas.

O capítulo está dividido em 2 secções:

1. Ferramentas;
2. Procedimentos.

A secção das ferramentas endereça o motor de jogo, a aplicação desktop de SIG e os scripts de Python utilizados, enquanto os procedimentos debruçam-se sobre a produção do mapa exterior em si.

Numa perspectiva de clareza, como as superfícies aleatórias são o ponto de partida e a base das operações de análise espacial, passarão a ser designadas “primitivas” a partir de agora.

Noutra nota, para centralizar o capítulo nos métodos, adoptou-se por uma abordagem adimensional e relativa. Quer isto dizer que, por exemplo, os valores da

temperatura não estão expressos em unidades conhecidas, e a reclassificação dos mesmos não é determinada através da predefinição de classes, mas sim classificada relativamente ao domínio dos valores da primitiva.

### **3.1 FERRAMENTAS**

Nativamente, as aplicações desktop de SIG não disponibilizam ferramentas de geração de superfícies aleatórias coerentes. No entanto, a possibilidade de programar ferramentas “à medida” está integrada através da utilização de linguagens de programação, como por exemplo: Python.

Nas subsecções abaixo estão apresentados o sistema de criação de jogos, a aplicação desktop de SIG: ArcGIS 10.1; bem como os scripts criados para possibilitar a geração das primitivas.

#### **3.1.1 MOTOR DE JOGO**

Para testar os mapas produzidos foi utilizado o *software* Unity 4.5.4

Unity é um sistema de criação de videojogos que inclui um motor de jogo e ambiente de desenvolvimento integrado, desenvolvido pela Unity Technologies. Para além de estar disponível em vários sistemas operativos (*e.g.* Mac OS, Linux, Windows) possibilita o desenvolvimento de jogos para múltiplas plataformas, incluindo consolas, sendo o SDK (*Software Development Kit*) por omissão para a Nintendo WiiU (MCELROY, 2013)

#### **3.1.2 APLICAÇÃO DESKTOP DE SIG**

A aplicação escolhida para o desenvolvimento do trabalho foi o ArcGIS 10.1.

ArcGIS (ESRI, 2012) é um *software* comercial de SIG que permite a criação, visualização, manipulação, análise e gestão de informação geográfica.

Um dos critérios da escolha foi o acesso imediato às operações já apresentadas através da licença para a extensão Spatial Analyst.

Outro critério na escolha, foi a integração da linguagem de programação Python (a versão 10.1 do ArcGIS vem complementada com a versão 2.7 de Python).

### 3.1.3 EXTENSÕES PYTHON

Optou-se pela utilização de 2 scripts para possibilitar a geração das primitivas via deslocamento do ponto central e ruído Perlin.

A primeira questão que se coloca é: programar de raiz, ou adoptar soluções já estabelecidas?

O caminho escolhido foi o da adopção de soluções já estabelecidas, se bem que de formas distintas para cada um dos métodos:

- **Deslocamento do ponto central:** a função para a aplicação deste método foi adaptada do código de STOLLE (2011);
- **Ruído Perlin:** encontram-se disponíveis várias extensões com a implementação de funções de ruído, bem como funções de movimento Browniano fraccional, sendo as que as mais notáveis são:
  - Noise 1.2.1: <https://pypi.python.org/pypi/noise/>
  - Noiselib 1.5.1: <https://pypi.python.org/pypi/noiselib/> (Versão portada para python da versão c++: <http://libnoise.sourceforge.net/> )
  - The Python Computer Graphics Kit (cgkit): <http://cgkit.sourceforge.net/>

A extensão utilizada no decorrer do trabalho foi o “Python Computer Graphics Kit”, porque não só cumpre os requisitos relativos às funções que se pretende utilizar, como também disponibiliza documentação bastante clara e, possibilita instalação “one click”. Este último ponto facilita a replicação noutros sistemas, já que evita

problemas que possam estar relacionados com dependências de ficheiros ou bibliotecas não compatíveis.

A 2ª questão que se coloca é relativa ao formato das superfícies a criar. Uma vez que se tratam de superfícies contínuas o formato mais adequado é raster/matricial, e por questões de escrita, leitura e interpretação optou-se por gerar as mesmas através da criação de um ficheiro ASCII Raster da Esri.

Os detalhes deste formato podem ser consultados no Resource Center da Esri sob as categorias Geodata>Data Types>Raster and images. De uma forma geral o formato consiste num ficheiro de texto composto por um cabeçalho, que define as características da matriz, seguido dos valores reais das células separados por espaços, e tem o seguinte aspecto:

```
NCOLS xxx
NROWS xxx
XLLCORNER xxx
YLLCORNER xxx
CELLSIZE xxx
NODATA_VALUE xxx
Linha 1
Linha 2
...
Linha n
```

Em que a informação no cabeçalho corresponde às descrições na tabela 5:

Parâmetro	Descrição
NCOLS	Número de colunas
NROWS	Número de linhas
XLLCORNER	Coordenada x da origem
YLLCORNER	Coordenada y da origem
CELLSIZE	Dimensão da célula (em unidades do mapa)
NODATA_VALUE	Valor por omissão (-9999)

Tabela 5: Descrição dos parâmetros do cabeçalho de um ficheiro ASCII Raster (ESRI, 2012)

## 3.2 PROCEDIMENTOS

Nesta secção serão expostos os passos a tomar para a geração procedimental dos mapas, assumindo a seguinte estrutura:

1. Primitivas;
2. Modelo geral de aplicação;
3. Altimetria;
4. Rede hidrográfica;
5. Temperatura;
6. Humidade;
7. Biomas;
8. Importação para o motor de jogo.

Em primeiro lugar apresentam-se os scripts que possibilitam a criação das superfícies primitivas, seguidos da visão global sobre o modelo geral de aplicação.

As secções seguintes apresentam os detalhes dos diferentes passos do modelo, e no final é descrito o processo de importação dos mapas para o motor de jogo.

### 3.2.1 PRIMITIVAS

O primeiro passo para iniciar a criação dos mapas é a geração das superfícies primitivas. Para o efeito, foram escritos dois scripts em Python:

Um para aplicar a função de ruído Perlin e movimento fraccional browniano, e outro para aplicar um algoritmo de deslocamento do ponto central.

Os detalhes de cada um estão apresentados abaixo em secções próprias.

### 3.2.1.1 RUÍDO PERLIN (fBM)

O script que gera primitivas baseadas em ruído foi escrito com auxílio à extensão “Python Computer Graphics Kit”, através da importação da função:

$$fBm((x, y, z), oitavas, lacunaridade, ganho) \quad (6)$$

Em que os argumentos são aqueles já descritos no enquadramento teórico, e devolve um valor no intervalo [0;1].

O pseudo-código utilizado para gerar a superfície (raster) foi o seguinte:

```
para cada x no número de colunas:  
  para cada y no número de linhas:  
    escreve fBm((x,y,0), oitavas, lacunaridade, ganho)
```

Em termos de inputs do script em si optou-se pelas seguintes variáveis (caixa de diálogo na figura 24):

- **Dimensão:** para simplificar o exercício o script vai gerar áreas quadradas, e esta variável determina o número de células do lado do quadrado (o tamanho da célula ficou omissa, com o valor de 25 unidades);
- **Período:** variável de controlo, com justificação detalhada a seguir à enumeração;
- **Oitavas:** número de iterações da fBm, tal como visto no enquadramento;
- **Amplitude:** valor que afecta o domínio dos valores na superfície gerada. Este valor é multiplicado pelo resultado da função;
- **Nome do ficheiro:** nome do ficheiro a criar.

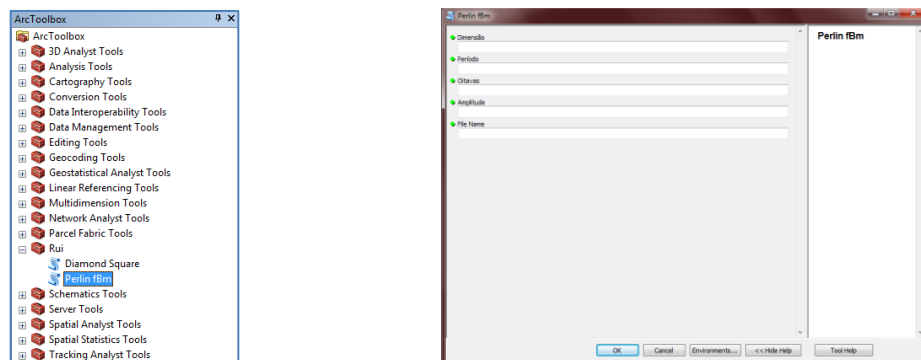


Figura 24: Localização do script Perlin fBm na toolbox e caixa de diálogo.



Pelas características da função, nas coordenadas inteiras o valor é constante. Logo, ao iterar no número de linhas e colunas (inteiros) a superfície seria constante. A variável período modifica a função da seguinte forma:

$$fBm\left(\left(\frac{x}{\text{período}}, \frac{y}{\text{período}}, z\right), \text{oitavas}, \text{lacunaridade}, \text{ganho}\right) \quad (7)$$

Garantindo a variabilidade da superfície e servindo também um segundo propósito: permite exemplificar superfícies para diferentes oitavas, correndo o script apenas para 1 oitava, variando manualmente este parâmetro e a amplitude. A denominação de período resulta do comportamento da função: uma divisão por 2 deste parâmetro resulta na duplicação da frequência.

O código final do script encontra-se no Anexo 1.

### 3.2.1.2 DESLOCAMENTO DO PONTO CENTRAL

O código para este script foi adaptado do código de STOLLE (2011) e corresponde ao algoritmo Diamante-quadrado apresentado no enquadramento.

Em termos de pseudo-código, a abordagem é a seguinte:

```
Enquanto lado do quadrado > 0:
    passo diamante;
    passo quadrado;
    aresta = aresta/2;
    chama a própria função 4x #uma para cada novo quadrado.
```

A função implementada recebe 8 argumentos:

$$\text{diamondsquare}(x, y, \text{comprimento}, \text{altura}, c_1, c_2, c_3, c_4) \quad (8)$$

Em que

- $x$  e  $y$  correspondem às coordenadas da vértice inferior esquerdo do quadrado;
- $\text{comprimento}$  e  $\text{altura}$  correspondem às coordenadas do vértice superior direito do quadrado;

- $c_i$  são os valores iniciais nos vértices do quadrado que corresponde à área total.

Quanto às variáveis que o script aceita como input (figura 25):

- **Dimensão:** número de células do lado da área quadrada;
- **Tamanho da célula:** tamanho de cada célula;
- **Amplitude:** valor que afecta a amplitude de cada valor aleatório somado ao valor do ponto central;
- **Nome do Ficheiro:** nome do ficheiro a criar.

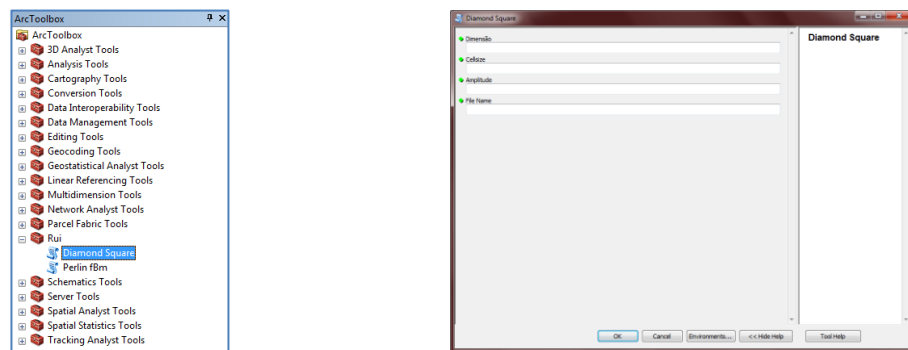


Figura 25: Localização do script Perlin fBm na toolbox e caixa de diálogo.

Ao contrário do script anterior, a área desta superfície tem que necessariamente ser quadrada, e potência de 2 pela forma como o algoritmo aplica as divisões sucessivas.

Por omissão, a função é chamada da seguinte forma:

$$\text{diamondsquare}(0,0,\text{dimensão},\text{dimensão},0,0,0,0) \quad (9)$$

A escolha inicial do valor 0 para todos os vértices é intencional, porque permite mais facilmente a criação de ilhas.

O código encontra-se no Anexo 2.

### 3.2.2 MODELO GERAL DE APLICAÇÃO

O esquema do modelo geral de aplicação (figura 26) apresenta os passos tomados com recurso ao *software* de SIG:

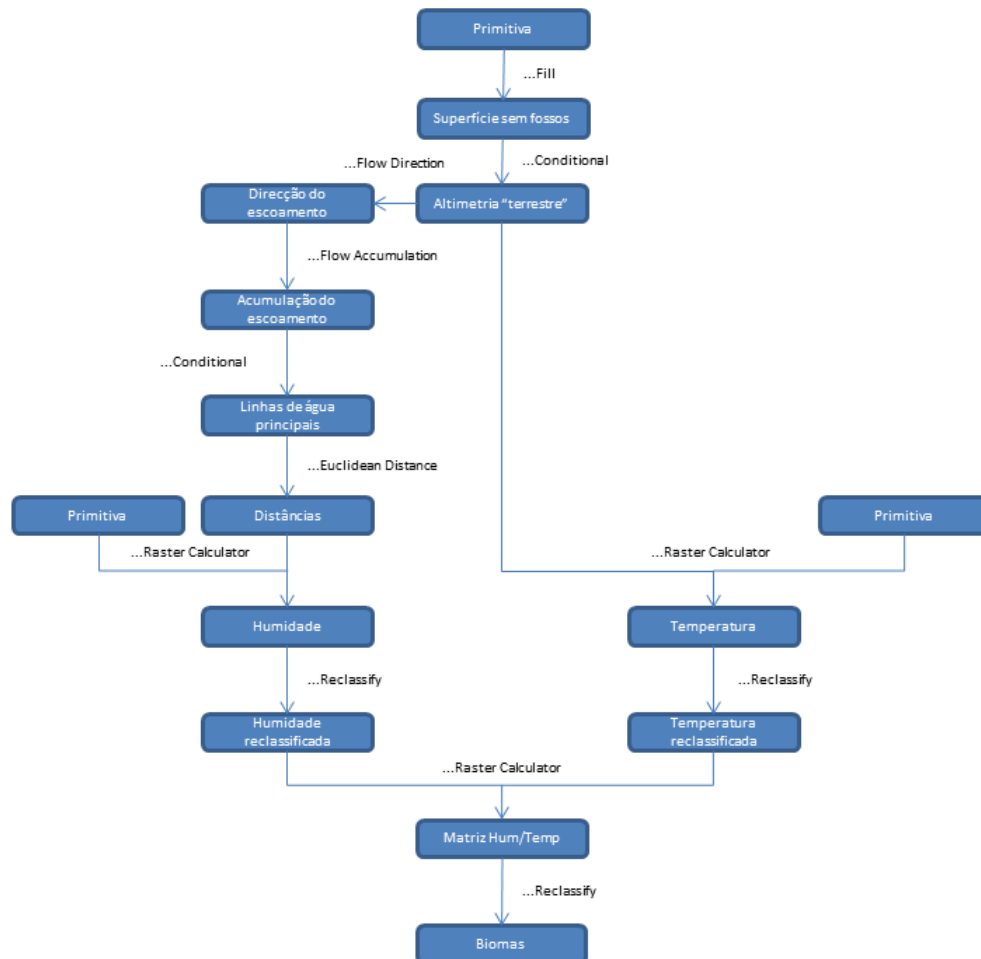


Figura 26: Modelo geral de aplicação.

Este modelo corresponde à emulação, com recurso ao *software* de SIG, da metodologia usual de geração procedimental de mapas em videojogos. Em primeiro lugar é gerada a altimetria, da qual são derivados os mapas de humidade e temperatura (neste caso concreto) que são sobrepostos para gerar os biomas.

Os passos estão detalhados nas secções que se seguem.

### 3.2.3 ALTIMETRIA

Nesta abordagem, a altimetria é o primeiro mapa concreto a gerar. De modo a preparar a superfície para as análises posteriores, são aplicadas as operações na figura 27:

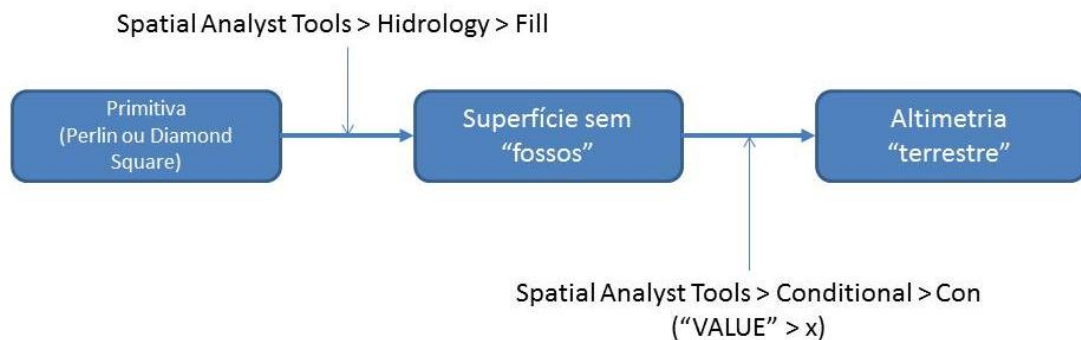


Figura 27: Esquema da geração da altimetria.

Começa-se pela geração de uma primitiva (via um dos scripts) e é necessário remover os picos e áreas de drenagem interna através da operação *Fill* já descrita no capítulo do enquadramento. Para além de preparar a superfície para a modelação hidrológica, o preenchimento das áreas de drenagem interna ajuda a definir áreas relativamente homogéneas, quando comparadas com a primitiva que é necessariamente mais rugosa.

A aplicação da operação Condicional vai definir as linhas de costa arbitrando um valor  $x$ , acima do qual se mantém o valor da superfície e abaixo do qual se reescreve como valor nulo (também a pensar no passo da modelação hidrológica).

### 3.2.4 REDE HIDROGRÁFICA

Uma vez gerada a altimetria, está-se em condições de aplicar as operações de direcção e acumulação do escoamento, tal como descritas no enquadramento, segundo o esquema da figura 28:

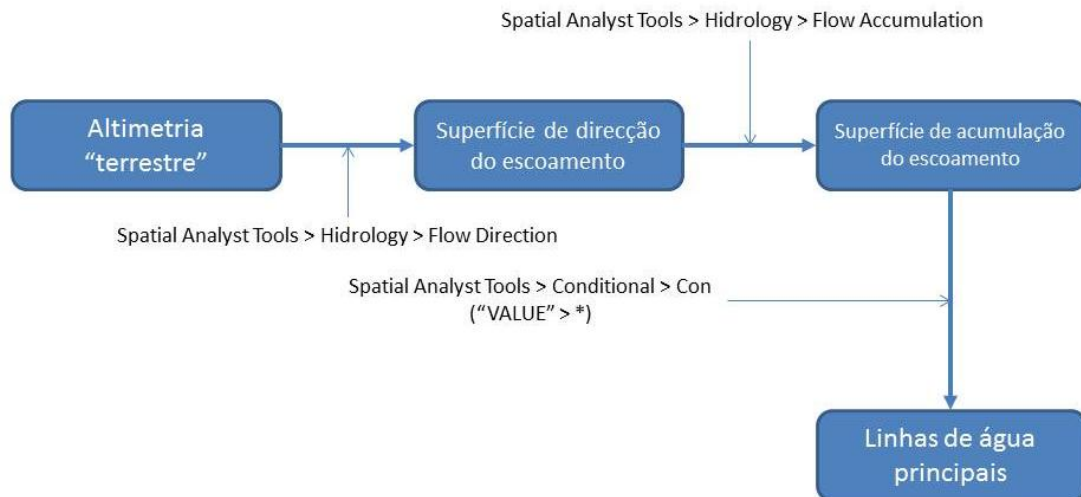


Figura 28: Esquema da geração da rede hidrográfica.

A última operação permite discriminar as células que identificam a rede hidrográfica, definindo um limiar na acumulação a partir do qual se considera que a célula representa uma linha de água. É colocado um \* no esquema acima, porque o racional para a determinação desse limiar depende do valor máximo de acumulação registado. Assumiu-se um valor que corresponde a 10% do valor máximo registado para a acumulação.

Esta relação empírica foi determinada com base numa abordagem “Gerar e Testar”, em que o modelo geral foi aplicado para vários limiares de acumulação, expressos em percentagem da acumulação máxima registada. O valor foi escolhido essencialmente através de critérios visuais, quer da densidade de linhas de água resultantes, quer da variabilidade dos biomas produzidos.

### 3.2.5 TEMPERATURA

De forma a convencionar uma superfície de temperatura relativamente intuitiva, optou-se por gerar uma primitiva à qual é subtraído o valor da altimetria (figura 29). Este processo vai estabelecer a relação (a menos da componente aleatória) de que a maiores altitudes, menor a temperatura.

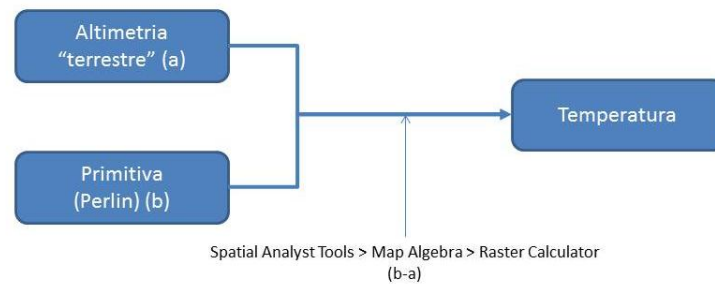


Figura 29: Esquema da geração da temperatura.

### 3.2.6 HUMIDADE

A denominação de “humidade” desta variável não é a utilização mais correcta. O termo é empregue por proximidade e simplificação do conceito. O que esta variável comporta é a distância às linhas de água principais (identificadas no procedimento da rede hidrográfica), modificadas por uma primitiva (figura 30).

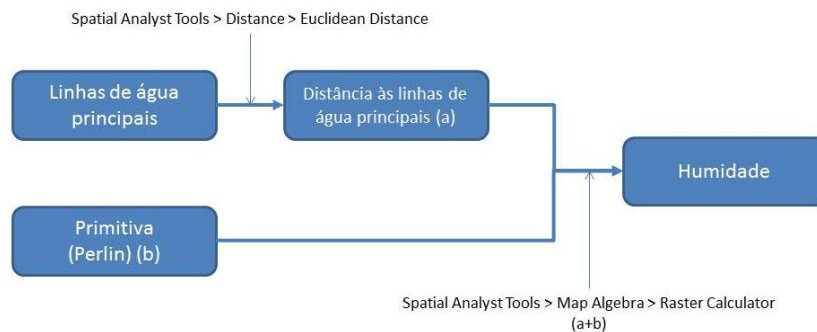


Figura 30: Esquema da geração da humidade.

A operação *Euclidean Distance* produz um raster em que cada célula comporta o cálculo da menor distância às células não nulas do raster de origem, neste caso as linhas de água principais.

Pelo domínio da superfície de distâncias (a), a primitiva (ruído) tem que ser gerada necessariamente com amplitude superior a 1. Arbitra-se para este caso uma amplitude que corresponda a 50% do valor máximo verificado na superfície de distâncias. À semelhança do limiar de acumulação na determinação da rede hidrográfica, esta relação também foi determinada empiricamente por análise dos biomas gerados, face à variação da amplitude da primitiva.

### 3.2.7 BIOMAS

Para finalizar o ensaio, utilizam-se as superfícies de temperatura e humidade para determinar os biomas segundo um diagrama de WHITTAKER (1975) adaptado (tabela 6), utilizando a humidade como *proxy* da precipitação.

		Temperatura				
		1	2	3	4	5
Humidade	10	Terras Estéreis	Deserto Temperado			Deserto
	20	Tundra		Pradaria		
	30					
	40	Neve	Matagal	Floresta Temperada Caducifolia		Floresta Tropical Caducifolia
	50		Taiga	Floresta Perenifolia	Floresta Temperada	Floresta Tropical Perenifolia

Tabela 6: Matriz da adaptação do diagrama de WHITTAKER (1975).

A aplicação desta matriz pressupõe a reclassificação das superfícies de temperatura e humidade em 5 classes, através do método das quebras naturais (*Natural Breaks*) (DE SMITH *et al.*, 2009):

- As classes de temperatura estão codificadas de 1 (temperatura mais baixa) a 5 (temperatura mais alta);
- As classes de humidade estão codificadas de 10 (humidade mais baixa) a 50 (humidade mais elevada)

A diferença na ordem de grandeza reside no resultado que se pretende: desta forma, a soma das duas superfícies vai resultar na codificação de cada entrada da matriz. Por exemplo, o resultado 33 corresponde unicamente à soma da classe de humidade 30 e de temperatura 3, que representa o bioma Matagal.

Assim sendo, o esquema da geração de biomas é dado pela figura 31.

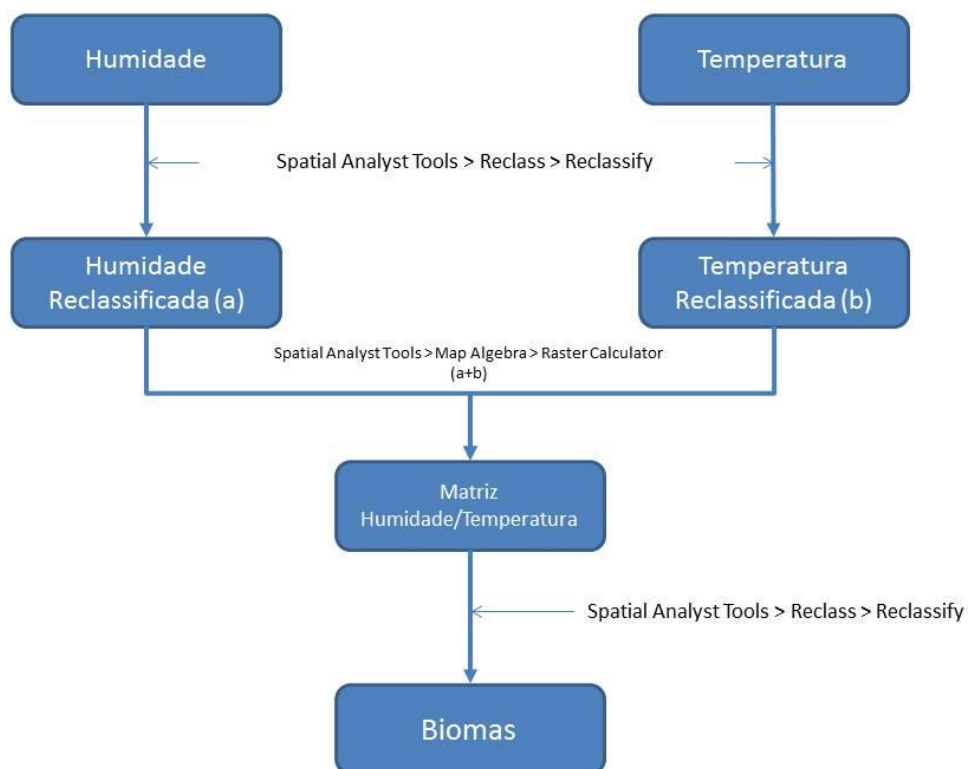


Figura 31: Esquema da geração de biomas.



### 3.2.8 IMPORTAÇÃO PARA O MOTOR DE JOGO

A possibilidade de leitura de formatos geográficos não é uma funcionalidade presente neste *software*, nem noutros do mesmo género (MAT *et al.*, 2014). A importação de modelos digitais do terreno no Unity é feita através de imagens no formato RAW (8bit ou 16bit), com dimensões iguais a  $2^n + 1$ , *i.e.*, 513x513, 1025x1025, 2049x2049 etc..

Esta limitação requer que o raster da altimetria deve ser exportado como imagem, e convertido posteriormente com recurso, por exemplo, à ferramenta *gdal\_translate* (<http://www.gdal.org>) através o seguinte comando:

```
gdal_translate -ot UInt16 -scale -of ENVI -outsize 1025 1025 altimetria.tif altimetria.raw
```

As opções utilizadas são as seguintes:

- **-ot UInt16:** utilização de um canal 16 bit.
- **-scale:** converte os valores do ficheiro original para o intervalo [0 - 65535].
- **-of ENVI:** define o binário RAW que é aceite pelo Unity.
- **-outsize 1025 1025:** o ficheiro deve ter dimensões iguais a  $2^n + 1$ .

Por outro lado, e para efeitos de visualização, o raster dos biomas pode ser exportado como imagem e importado no motor de jogo como textura.

A importação do ficheiro da altimetria é feita através da opção:

*Game Object > Create Other > Terrain > Heightmap > Import Raw.*

Estão apresentados nas secções abaixo os resultados para cada um dos procedimentos:

1. Primitivas;
2. Altimetria;
3. Rede hidrográfica;
4. Temperatura
5. Humidade
6. Biomas
7. Motor de Jogo

Os pontos 1 a 6 correspondem aos resultados produzidos na aplicação de SIG, e vão ser apresentados 2 mapas em cada um dos passos. A diferença é a primitiva que lhes deu origem. Por exemplo, a *Temperatura Perlin* será a Temperatura correspondente ao mapa originalmente gerado por ruído Perlin, e da mesma forma, *Rede Hidrográfica deslocamento do ponto central* corresponde à rede hidrográfica derivada de uma superfície originalmente gerada pelo script *Diamond Square*.

Nos casos em que o procedimento requer introdução de variáveis de entrada, as mesmas precedem a apresentação dos mapas sob a forma de tabela.

O último ponto (7) apresenta algumas imagens da importação da altimetria e biomas (deslocamento do ponto central) para o motor de jogo.

### 4.1 PRIMITIVAS

A tabela 7 contém os parâmetros utilizados em cada uma dos *scripts*:

<i>Script</i>	<i>Dimensão</i>	<i>Amplitude</i>	<i>Cellsize</i>	<i>Período</i>	<i>Oitavas</i>
<i>Perlin fBm</i>	1024	1	25 (omissão)	1024	5
<i>Diamond Square</i>	1024	1	25	N/A	N/A

Tabela 7: Parâmetros dos *scripts* para geração das primitivas.

As superfícies geradas por cada um dos *scripts* estão apresentadas na figura 32:

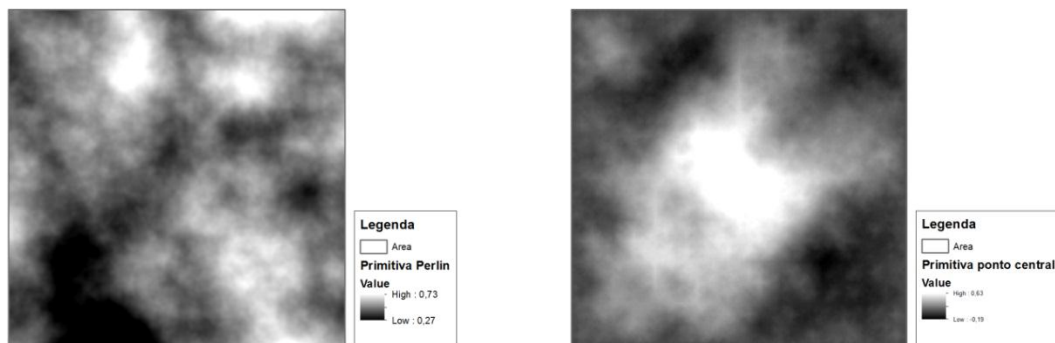


Figura 32: Primitivas – Perlin à esquerda, deslocamento do ponto central à direita.

## 4.2 ALTIMETRIA

A tabela 8 comporta os valores de corte assumidos na produção da altimetria “terrestre”.

Altimetria	“VALUE” >
Perlin	0.49
Deslocamento do ponto central	0.05

Tabela 8: Valor de corte da operação *Spatial Analyst Tools > Conditional > Con.*

Na figura 33 estão expostas as superfícies resultantes, por classes de altimetria:

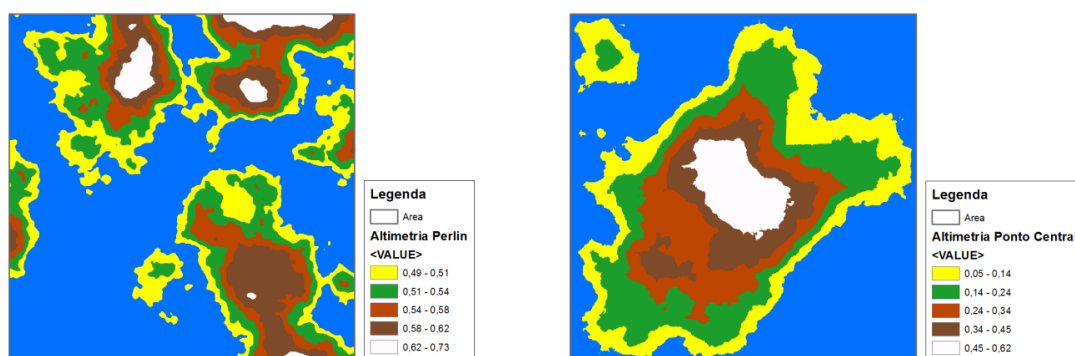


Figura 33: Altimetrias – Perlin à esquerda, deslocamento do ponto central à direita.

### 4.3 REDE HIDROGRÁFICA

Os valores assumidos para a acumulação do escoamento na definição das linhas de água principais encontram-se registados na tabela 9 abaixo:

Acumulação do escoamento	“VALUE” > (10% do escoamento máximo)
Perlin	3026
Deslocamento do ponto central	7363

Tabela 9: Acumulação do escoamento - valor de corte da operação *Spatial Analyst Tools* >Conditional > Con.

Na figura 34 estão assinaladas as linhas de água principais:

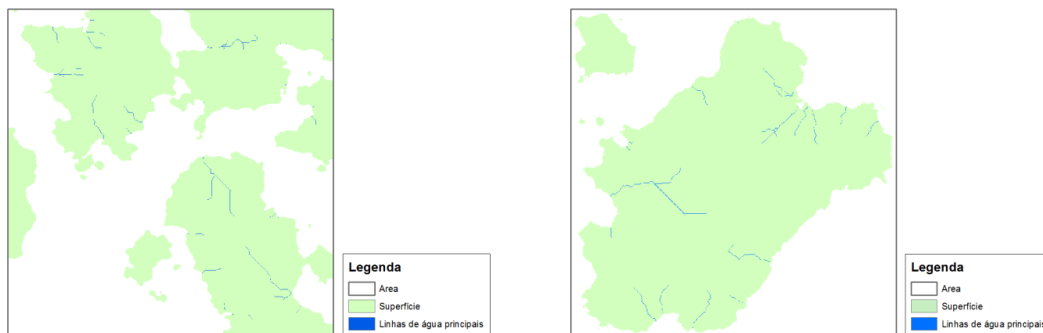


Figura 34: Rede hidrográfica – Perlin à esquerda, deslocamento do ponto central à direita.

### 4.4 TEMPERATURA

A determinação das superfícies de temperatura envolve a soma de uma componente aleatória determinada através da geração de ruído Perlin. Para ambos os casos as superfícies primitivas foram geradas com recurso ao script *Perlin fBm* com os

parâmetros presentes na tabela 10. As superfícies finais da Temperatura estão apresentadas na figura 35 já com a respectiva reclassificação.

<i>Script</i>	<i>Dimensão</i>	<i>Amplitude</i>	<i>Cellsize</i>	<i>Período</i>	<i>Oitavas</i>
<i>Perlin fBm</i>	1024	1	25 (omissão)	512	2

Tabela 10: Parâmetros do *script* para geração do ruído da temperatura.

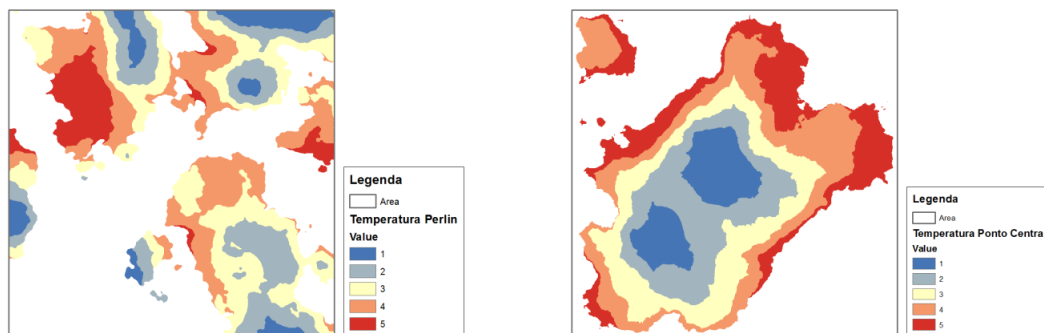


Figura 35: Temperaturas – Perlin à esquerda, deslocamento do ponto central à direita.

## 4.5 HUMIDADE

Os parâmetros utilizados no script Perlin fBm para a geração das componentes aleatórias no cálculo da Humidade estão listados na tabela 11:

	<i>Dimensão</i>	<i>Amplitude (50% da distância máxima)</i>	<i>Cellsize</i>	<i>Período</i>	<i>Oitavas</i>
Perlin	1024	8000	25 (omissão)	128	3
Deslocamento do ponto central	1024	5500	25 (omissão)	128	3

Tabela 11: Parâmetros dos *scripts* para geração do ruído da humidade.

A figura 36 mostra as superfícies resultantes de Humidade, já reclassificadas.

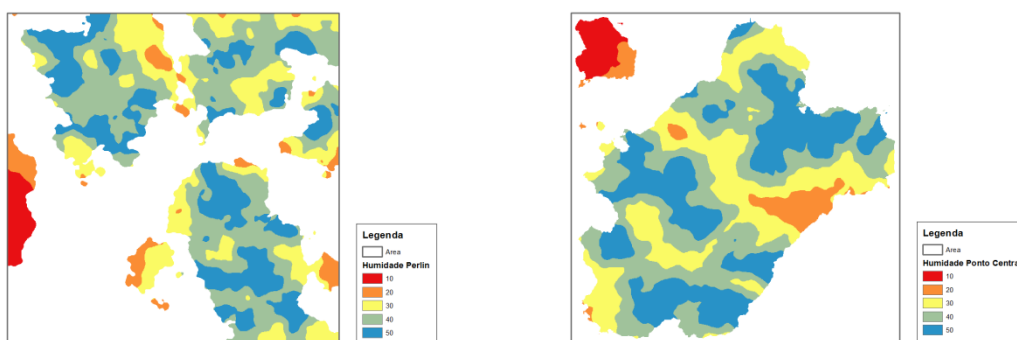


Figura 36: Humidades – Perlin à esquerda, deslocamento do ponto central à direita.

## 4.6 BIOMAS

O resultado final do modelo em SIG está presente na figura 37.

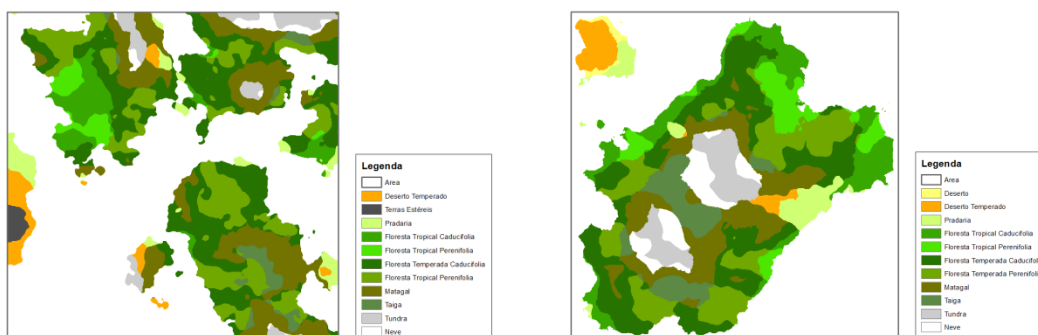


Figura 37: Biomias – Perlin à esquerda, deslocamento do ponto central à direita.

## 4.7 MOTOR DE JOGO

Na figura 38 estão duas capturas da altimetria e biomas correspondentes à superfície inicial gerada pelo algoritmo do deslocamento do ponto central.



---

Figura 38: À esquerda – vista geral do modelo do terreno com distribuição de biomas; à direita – modo de jogo com vista na primeira pessoa.

Este capítulo seguirá a estrutura do capítulo anterior, debruçando-se numa primeira fase sobre cada um dos procedimentos, para no final poder ser feita uma comparação entre os mapas criados.

### 5.1 PRIMITIVAS

Os resultados produzidos pelos scripts adoptados têm diferenças evidentes. Em primeiro lugar, os domínios dos valores das superfícies aleatórias são diferentes: uma primitiva gerada pelo script Perlin fBm tem sempre domínio  $[0; 1]$ , enquanto que pelo script Diamond Square, o domínio é variável.

Também os valores em cada uma das superfícies criadas têm distribuições diferentes: a superfície criada pelo script Perlin fBm tem uma distribuição normal, enquanto que no script Diamond Square a distribuição aproxima-se da linear (figura 39)

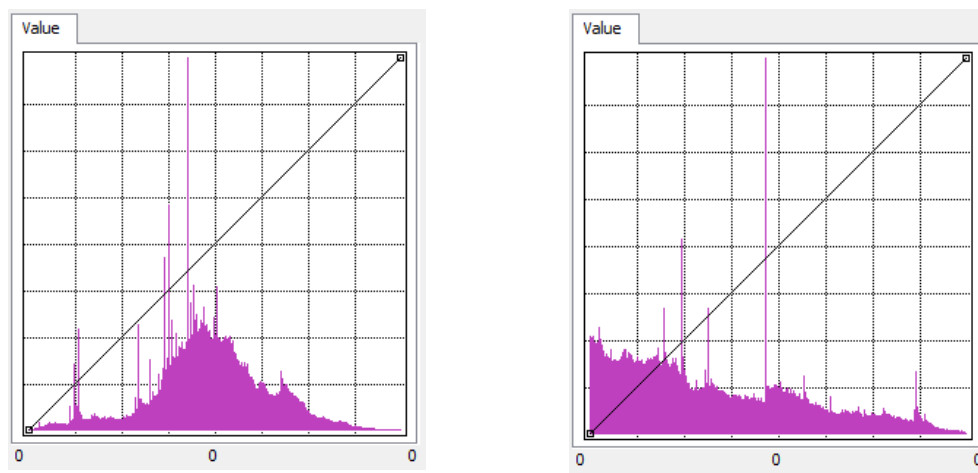
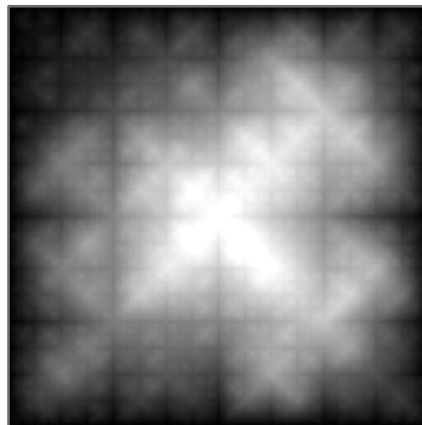


Figura 39: Distribuições – Perlin fBm à esquerda; Diamond Square à direita.

O algoritmo no script Perlin fBm, ao contrário do Diamond Square, não permite a definição de valores nos limites da área que se pretende gerar. Isto faz com que as superfícies criadas desta forma fiquem “cortadas” nos limites. Já com o script Diamond Square, uma vez que se considerou por omissão o valor 0 para as



coordenadas dos cantos, no passo seguinte (definição da altimetria) é possível ter a formação de ilhas (caso na primeira iteração a componente aleatória seja positiva) ou lagos (caso a componente aleatória seja negativa). Uma das desvantagens deste último método é não conseguir prever se o resultado vai ser uma ilha, ou um lago. Isto acontece porque na aplicação do algoritmo, a componente aleatória é gerada no intervalo  $[-1;1]$ . A alternativa para forçar a formação de ilhas seria colocar a componente aleatória no intervalo  $[0; 1]$ , mas os resultados (exemplo na figura 40) apenas evidenciam ainda mais outra das desvantagens do algoritmo: é possível identificar visualmente os vestígios das iterações (passo diamante e passo quadrado) ao longo dos eixos vertical, horizontal e nas diagonais. Esta questão não é tão evidente com uma componente aleatória no intervalo  $[-1; 1]$



---

Figura 40: Vestígios da aplicação do algoritmo de deslocamento do ponto central com componente aleatória no intervalo  $[0; 1]$ .

Relativamente ao script Perlin fBm, dependendo da finalidade, é pertinente experimentar várias combinações nos parâmetros de entrada, particularmente nas variáveis período e oitavas. Quanto mais pequeno for o período relativamente à dimensão, maior a variabilidade da superfície, e quanto maior o número de oitavas, maior é o “detalhe” do ruído. No limite, períodos pequenos e elevado número de oitavas aproximam-se de ruído branco.

## 5.2 ALTIMETRIA

A altimetria é o produto mais directamente relacionado com as primitivas, e portanto “herda” os aspectos referidos na secção anterior.

Há, no entanto, 3 questões pertinentes na geração da altimetria que são válidas para ambos os métodos:

1. opção de produzir uma altimetria “terrestre”, em que apenas se consideram valores acima de determinado limiar;
2. operação *Fill* para eliminar áreas de drenagem interna;
3. continuidade e variabilidade da superfície impedem formações geomorfológicas específicas/individualizadas.

Os dois primeiros pontos são opções tomadas a pensar no passo da geração da rede hidrográfica. “Cortar” as primitivas abaixo de determinado valor tem a desvantagem de se perder a forma das superfícies submersas, mas em contrapartida, ao reduzir a área que vai contribuir para a acumulação do escoamento, permite a definição de uma rede hidrográfica menos densa a culminar nas linhas de costa definidas pelo corte. A operação *Fill* é necessária para assegurar a continuidade e aplicação correcta do algoritmo de acumulação do escoamento, mas impossibilita a criação de lagos. Esta última limitação não é totalmente negativa, uma vez que o preenchimento das áreas de drenagem interna possibilita a homogeneização de algumas zonas, o que introduz alguma diversidade numa superfície que doutra forma é “regularmente irregular”.

O terceiro ponto também toca na questão da superfície ser “regularmente irregular”. O facto de não haver descontinuidades significativas nas superfícies faz com que, de uma forma geral, toda a variação se acabe por tornar monótona porque não sobressaem elementos geomorfológicos distintos.

Há alternativas a considerar em trabalhos futuros para ultrapassar este último ponto, como por exemplo:

- modificar o algoritmo de ruído Perlin, de forma a produzir valores segundo outras funções de distribuição (PARBERRY, 2014);

- utilizar uma abordagem inicial vectorial, por exemplo como PATEL (2010), onde os polígonos de Voronoi aleatórios podem garantir homogeneidade entre polígonos da mesma classe e heterogeneidade entre polígonos de classes distintas, utilizando funções de distribuição diferentes para cada classe.

### 5.3 REDE HIDROGRÁFICA

Como foi referido na secção 3.2, as linhas de água principais foram determinadas através da selecção das células cuja acumulação do escoamento fosse superior a 10% do valor máximo de acumulação registado. Este valor foi escolhido com base no gráfico apresentado na figura 41, em se pode ver a influência do valor de corte (percentagem em função do valor máximo de acumulação) na variação dos biomas, no qual se considerou que 10% correspondia a uma distribuição aceitável destes últimos.

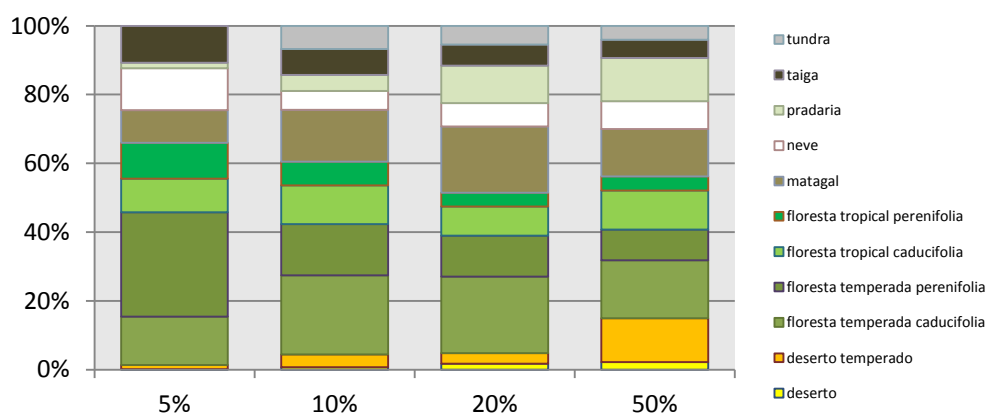
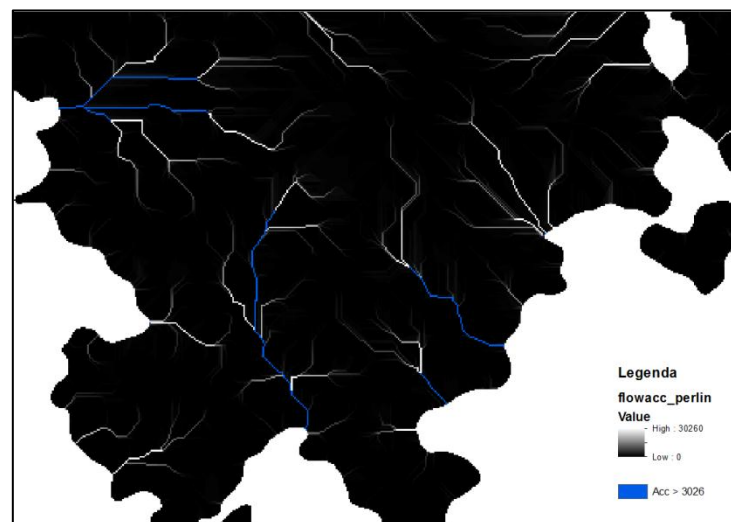


Figura 41: Variação da cobertura dos biomas face à variação do limiar de acumulação do escoamento (em função do escoamento máximo verificado).

Na figura 42 abaixo pode comparar-se o resultado das linhas de água principais (a azul, que correspondem a uma acumulação superior a 10% da acumulação máxima), em relação a toda a superfície de acumulação. A utilização de valores de corte menores torna a rede hidrográfica mais densa, e no limite, igual à superfície de

acumulação. O inverso reduz a densidade da rede e conduz a linhas de água principais de menos comprimento.

Como a variável “Humidade” utilizada na classificação dos biomas depende directamente destes resultados, é possível ver no gráfico da figura 41 que à medida que se aumenta o valor de corte e se reduz a densidade das linhas água principais, os desertos e a pradaria ganham expressão, às custas da redução das áreas classificadas como florestas.



---

Figura 42: Pormenor da acumulação do escoamento na altimetria Perlin.

## 5.4 TEMPERATURA

A geração da temperatura foi generalizada simplesmente como “função” da altimetria. Introduziu-se a componente aleatória para que a temperatura não fosse apenas inversamente proporcional à altimetria, mas procurou garantir-se que a regra geral fosse: temperaturas mais baixas ocorrem a altitudes mais elevadas. Os resultados mostram que esta relação foi conseguida.

## 5.5 HUMIDADE

A humidade foi utilizada com o intuito de servir de *proxy* à precipitação para a aplicação do diagrama adaptado de WHITTAKER (1975), e a principal questão levantada na geração desta variável, foi a definição da amplitude da componente aleatória.

A abordagem foi semelhante à escolha do valor de corte para a acumulação do escoamento. Foram calculadas as percentagens de cobertura dos diferentes biomas no mapa final para diferentes amplitudes da primitiva Perlin utilizada no ruído (expressas em função da distância máxima verificada). Pelo resultado no gráfico da figura 43, as percentagens de cobertura dos biomas não aparentam ter variações significativas para diferentes amplitudes.

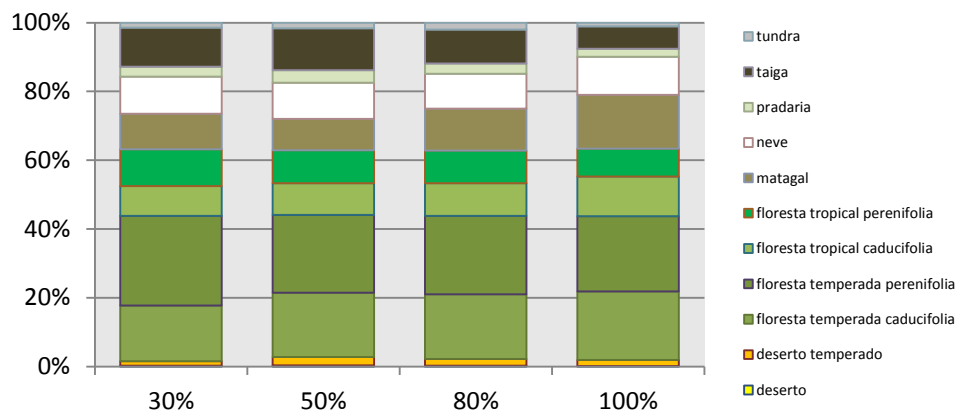


Figura 43: Variação da cobertura dos biomas face à variação da amplitude do script de ruído (em função da distância máxima às linhas de água).

As diferenças verificam-se essencialmente na forma da superfície, onde menores valores na amplitude do ruído geram formas que se assemelham a *buffers* (o equivalente a não ter ruído) enquanto para valores mais elevados têm-se maior variabilidade nas formas (figura 44).

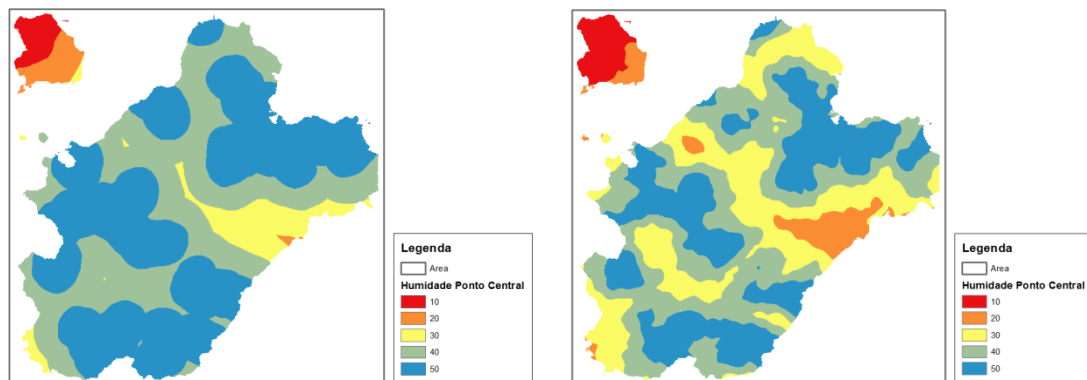


Figura 44: Exemplo da humidade do deslocamento do ponto central: resultado com amplitude do ruído igual a 10% da distância máxima à esquerda; 50% à direita.

## 5.6 BIOMAS

A geração dos biomas é a operação mais simples em termos de aplicação: consiste apenas na soma de duas variáveis e reclassificação segundo uma matriz. No entanto, é neste passo que se reflectem todas as generalizações e simplificações consideradas no modelo, mais concretamente no que diz respeito ao método de reclassificação das variáveis temperatura e humidade.

A reclassificação pelo método das quebras naturais apresenta sempre o número de classes necessário, neste caso 5 classes quer para a temperatura quer para a humidade, para aplicar a matriz dos biomas. Isto pode ser simultaneamente uma vantagem e uma desvantagem. Se o objectivo for produzir sempre uma grande diversidade de biomas, então pode ser visto como vantagem, mas se quisermos ter a possibilidade de mapas dominados por determinado conjunto de biomas, então este aspecto já não é positivo. Outra limitação deste método de classificação é que é específico da informação sobre o qual opera (DE SMITH *et al.*, 2009). Isto quer dizer que não é muito útil na comparação entre múltiplos mapas, cuja informação subjacente tem domínios diferentes. Assim, não é muito correcto comparar uma reclassificação de uma temperatura com base em primitivas Perlin com outra com base no deslocamento do ponto central.

Do ponto de vista do exercício, a geração de um mapa com praticamente todos os biomas é um aspecto positivo. Para além do mais, neste cenário, para obter áreas dominadas por um conjunto limitado de biomas uma alternativa é gerar uma primitiva tão grande quanto o necessário para seleccionar apenas no final uma área que satisfaça o critério.

## 5.7 MOTOR DE JOGO

A interoperabilidade entre os formatos de dados geográficos e os formatos utilizados no *software* de desenvolvimento de jogos é um obstáculo. A menos de alguma aplicação ou extensão programada para o efeito, não é possível a importação directa de dados numa base de dados geográfica, ou outros formatos standard. Esta limitação é transversal aos principais motores de jogo no mercado (MAT *et al.*, 2014).

Em acréscimo, a conversão para os formatos importáveis no motor de jogo obriga à perda de informação (a conversão para o formato RAW 16 bit escala os valores para o intervalo [0 - 65535]), sendo necessário fazer o ajuste *a posteriori*.

Este último ponto não influencia em nada a metodologia seguida, uma vez que se decidiu trabalhar com unidades adimensionais.

Uma vez importada a altimetria é possível aplicar uma imagem exportada do mapa dos biomas como textura, um plano de água, e os resultados podem ser vistos na figura 38 (secção 4.7).

O mapa produzido é navegável, mas a aplicação dos biomas com textura apenas serve para enquadramento visual. As operações seguintes necessárias para a criação de um mapa mais credível não estão previstas no âmbito deste trabalho, mas passam pela aplicação correcta de texturas e distribuição de objectos (e.g. vegetação), segundo os respectivos biomas. Esta não é uma funcionalidade de origem do Unity, e tem de ser conseguida através de um *script* escrito para o propósito.

## 5.8 CONSIDERAÇÕES GERAIS

Como os resultados mostraram, é possível criar mapas exteriores aleatórios para videojogos com recurso a *software* de SIG. Há no entanto aspectos a referir que saem do enquadramento das secções anteriores.

Uma externalidade positiva que surge da funcionalidade de geração de superfícies aleatórias, é a utilização das mesmas para testar modelos e algoritmos, à semelhança do que foi feito no procedimento de determinação da rede hidrográfica. A possibilidade de gerar superfícies coerentes permite testar a robustez e sensibilidade de modelos quer aplicando o modelo a  $n$  superfícies diferentes, quer utilizando as superfícies aleatórias como superfícies de erro.

Não foi focado o aspecto da performance dos scripts em termos de tempos de execução, mas é possível avançar que mesmos não são competitivos nem comparáveis relativamente a outras aplicações. Considerando apenas a geração das superfícies, e tendo também em conta o tempo de escrita no ficheiro, os tempos podem chegar aos 10 segundos numa máquina com um processador Dual Core 2.53GHz e 4GB de RAM, enquanto que *software* específico como *World Machine*<sup>13</sup> na mesma máquina obtêm-se resultados na ordem dos milissegundos, permitindo previsualizar os efeitos da variação de parâmetros em tempo real.

A vantagem que esta metodologia tem relativamente ao *software* de geração procedimental, é que estes últimos focam-se essencialmente nos processos de modelação digital do terreno, enquanto que na abordagem SIG há a possibilidade de estender as análises para além da modelação digital do terreno. Por exemplo, análise potencial das áreas para a instalação de uma cidade. Este exercício não foi feito, mas derivando os declives e a orientação das encostas do modelo digital do terreno, em conjugação com as superfícies de temperatura e humidade, está-se em condições de definir um conjunto de critérios que permitam identificar as melhores áreas para o efeito.

---

<sup>13</sup> <http://www.world-machine.com/>



Considera-se que a estrutura do trabalho permitiu o cumprimento dos objectivos propostos no início.

Quanto à identificação dos pontos de convergência entre as tecnologias, dos 5 pontos assinalados por AHLQVIST (2011), a metodologia seguida no trabalho incidiu sobre os aspectos referentes à Análise Espacial e Construção de Mundos

Foi possível reproduzir, com recurso a *software* de informação geográfica, um procedimento de geração de mapas aleatórios que é comum no desenvolvimento de videojogos, e posterior importação para um motor de jogo. Este procedimento permitiu o seguinte:

- adaptação e integração em SIG de algoritmos geradores de superfícies aleatórias;
- aplicação de operações de análise espacial disponibilizadas no *software* de SIG para gerar e derivar nova informação que contribui para a construção de mundos;
- encontro de um dos maiores obstáculos à utilização em conjunto das duas tecnologias – a incompatibilidade de formatos.

Através destes 3 pontos é possível apontar algumas mais-valias e limitações da utilização de SIG na geração de mapas aleatórios.

No que diz respeito à integração de algoritmos geradores de superfícies aleatórias, estes são uma boa adição ao conjunto de ferramentas no *software* de SIG, mas para o objectivo concreto não oferecem o mesmo nível de desempenho e flexibilidade que outras soluções dedicadas, como o *software Terragen*<sup>14</sup> ou *Wold Machine*, que geram modelos digitais do terreno aleatórios com elevado nível de parametrização e pré-visualização em tempo real.

A vantagem da utilização de um SIG é que as funcionalidades analíticas permitem um nível mais aprofundado de modelação geográfica (permitem a modelação de

---

<sup>14</sup> <http://planetside.co.uk/>

variáveis ou processos para além dos modelos digitais de terreno) e a comparação objectiva entre resultados ou métodos. Outras vantagens prendem-se com a utilização de bases de dados geográficas para a geração de paisagens aleatórias segundo os perfis de paisagens conhecidas (PARBERRY, 2013, PARBERRY 2014), mas esta possibilidade não foi abordada no trabalho.

Quanto à incompatibilidade de formatos, o modelo genérico actual consiste em 3 passos: Exportação > Conversão > Importação. Um primeiro avanço para ultrapassar este obstáculo seria eliminar o passo da conversão, mas idealmente, a interoperabilidade deveria permitir o acesso “natural” aos dados nos dois sentidos. Como foi visto (secção 2.4), a tecnologia dos videojogos é mais volátil que a tecnologia de SIG, pelo que o que faz sentido seria o desenvolvimento do lado do *software* de videojogos de capacidades para lidar com os formatos standard de informação geográfica.

Sendo assim, da parte de SIG, podem propor-se as seguintes direcções para trabalhos futuros:

- Explorar a vertente de geração aleatória com recurso a um modelo de dados vectorial, passando por exemplo pela geração de polígonos de Voronoi aleatórios como PATEL (2010);
- Utilizar uma abordagem como PARBERRY (2013, 2014) e determinar as distribuições e perfis tipo de altimetria para um conjunto conhecido de paisagens, e modificar os algoritmos de geração para produzir paisagens aleatórias com as mesmas características.

Em suma, é possível utilizar um SIG para gerar mapas exteriores para videojogos mas, face às alternativas dedicadas para a geração procedimental de modelos digitais de terreno e tendo em conta que o *software* de SIG não tem uma finalidade criativa, não se recomenda a sua utilização nessa vertente. As maiores vantagens podem usufruídas no âmbito da análise espacial e modelação geográfica que ultrapassa o domínio dos modelos digitais do terreno, e ficam registadas duas possibilidades de desenvolvimentos futuros com o objectivo de aproximar os videojogos e SIG, de preferência pela adopção de metodologias padronizadas que beneficiem ambas as tecnologias.

- AHLQVIST, O., 2011, Converging Themes in Cartography and Computer Games, *Cartography and Geographic Information Science*, 38(3), pp. 278-285.
- ANGELIDES, M. e AGIUS, H., 2014, *Handbook of Digital Games*, Wiley-IEEE Press.
- ASHMORE, C., e NITSCHKE, M., 2007, The quest in a generated world, In *Proc. 2007 Digital Games Research Assoc.(DiGRA) Conference: Situated Play*, pp. 503-509.
- BELHADJ, F. e AUDIBERT, P., 2005, Modeling Landscapes with Ridges and Rivers: Bottom Up Approach. In *GRAPHITE '05: Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, pp. 447-450.
- BERRY, J., 2004, Bridging GIS and Map Analysis: Identifying and Utilizing Spatial Relationships, In *ASPRS Annual Conference*.
- BURROUGH, P., 1986, *Principles of geographical systems for land resources assessment*, Clarendon Press.
- CONCELMO, C., 2009, The Memory Card .62: Leaving Midgar, <http://www.destructoid.com/the-memory-card-62-leaving-midgar-142776.phtml>, consultado a 16 de setembro de 2014.
- COVEYOU, R., 1969, Random number generation is too important to be left to chance. *Applied Probability and Monte Carlo Methods and modern aspects of dynamics. Studies in applied mathematics*, 3, pp. 70-111.
- CRESSIE, N., 1993, *Statistics for Spatial Data*: Wiley Series in Probability and Statistics.
- CSIKSZENTMIHALYI, M., 1990, *Flow: The psychology of optimal experience*, Harper&Row.
- CULLEN, B. e O'SULLIVAN, C., 2011, A caching approach to real-time procedural generation of cities from GIS data, *Journal of WSCG*, pp. 119-126.

- DE SMITH, M., GOODCHILD, M., e LONGLEY, P., 2009, Geospatial analysis: a comprehensive guide to principles, techniques and software tools, 4<sup>a</sup> ed. <http://www.spatialanalysisonline.com/>, consultado a 07 de julho de 2014.
- DORMANS, J., 2011, Beyond iconic simulation, *Simulation & Gaming*, 42(5), pp. 610-631
- EBERT, D., MUSGRAVE, F., PEACHEY, D., PERLIN, K., e WORLEY, S., 2003, *Texturing and Modeling: A Procedural Approach*. Elsevier.
- EBNER, M., LEVINE, J., LUCAS, S. M., SCHAUL, T., THOMPSON, T., e TOGELIUS, J., 2013, Towards a video game description language, *Dagstuhl Follow-Ups*, 6.
- ESRI, 2012, ArcGIS Resource Center, <http://resources.arcgis.com/en/home/>, consultado a 8 de julho de 2014.
- FALCONER, A., e FORESMAN, J., 2002, *A system for Survival: GIS and Sustainable Development*, ESRI.
- FERNÁNDEZ-VARA, C., ZAGAL, J., e MATEAS, M., 2007, Evolution of Spatial Configurations in Videogames, *Worlds in play: International perspectives on digital games research*, 21, 159
- FOURNIER, A., FUSSELL, D., e CARPENTER, L., 1982, Computer rendering of stochastic models, *Communications of the ACM*, 25(6), pp. 371-384.
- FRANK, A., 1988, Requirements for a database management system for a GIS, *Photogrammetric Engineering & Remote Sensing*, 54(11), pp. 1557-1564.
- FRIESE, K., HERRLICH, M. e WOLTER, F., 2008, Using game engines for visualization in scientific applications, *In IFIP International Federation for Information Processing*, 279, *New Frontiers for Entertainment Computing*, pp. 11-22.
- FRITSCH, D., e KADA, M., 2004, Visualisation using game engines, *Archiwum ISPRS*, 35, B5.
- GOODCHILD, M., 1988, Geographic information systems, *Progress in Human Geography*, 12(4), pp. 560-566.

- GOODCHILD, M., 1992, Geographical information science, *International Journal of Geographical Information Systems*, 6(1), pp. 31-45.
- GOODCHILD, M. e KEMP, K. 1990. *NCGIA Core Curriculum: Introduction to GIS*.
- HASTINGS, E., GUHA, R. e STANLEY, K., 2009, Automatic content generation in the galactic arms race video game, *IEEE Transactions on Computational Intelligence and AI in Games*, 1(4), pp. 245-263.
- HENDRIKX, M., MELJER, S., VAN DER VELDEN, J., e IOSUP, A., 2013, Procedural content generation for games: A survey, *ACM Transactions on Multimedia Computing, Communications, and Applications*, 9(1), 1.
- HUIZINGA, J., 1956, *Homo ludens*, Hamburg: Rowohlt.
- IOSUP, A., 2011, POGGI: generating puzzle instances for online games on grid infrastructures, *Concurrency and Computation: Practice and Experience*, 23(2), pp. 158-171.
- JENSON, S. e DOMINGUE, J., 1988, Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric engineering and remote sensing*, 54(11), pp. 1593-1600.
- JUUL, J., 2005, *Half-real. Video Games between Real Rules and Fictional Worlds*, Cambridge.
- KANG, S., e KIM, S., 2014, Automated spatio-temporal analysis techniques for game environment. *Multimedia Tools and Applications*, 1-7.
- KELLEY, A, MALIN, M. e NIELSON, G., 1988, Terrain simulation using a model of stream erosion, *ACM SIGGRAPH Computer Graphics*, 22(4), pp. 263-268.
- KIM, A., 2012, The player's journey: designing over time, <http://amyjokim.com/2012/09/14/the-players-journey-designing-over-time/>, consultado a 22 de maio de 2014.
- KOSTER, R., 2013, *Theory of fun for game design*, O'Reilly Media, Inc.

- LI, Z., ZHU, C., e GOLD, C., 2010, *Digital terrain modeling: principles and methodology*, CRC Press.
- LONGLEY, P. GOODCHILD, M., MAGUIRE, D., e RHIND, D., 2005, *Geographical Information Systems and Science*, 2<sup>a</sup> ed. John Wiley & Sons.
- MAGUIRE, D., (1991). An overview and definition of GIS, In *Geographical Information Systems: principles and applications*, 1, pp. 9-20.
- MALIENE, V., GRIGONIS, V., PALEVIČIUS, V. e GRIFFITHS, S., 2011. Geographic information system: Old principles with new capabilities, *Urban Design International*, 16(1), pp. 1-6.
- MANDELBROT, B., 1979, *Fractals: form, chance and dimension*, WH Freeman & Co.
- MANDELBROT, B., 1983, *The fractal geometry of nature*, Henry Holt & Co.
- MARCZESWKI, A., 2012, Flow, player journey and employee satisfaction, <http://www.gamified.co.uk/2012/11/30/flow-and-satisfaction/>, consultado a 22 de maio 2014.
- MCELODY, G., 2013, Unity for Wii U opens up GamePad hardware and more to developers, <http://www.polygon.com/2013/8/20/4641786/unity-for-wii-u-opens-up-gamepad-hardware-and-more-to-developers>, consultado a 16 de setembro de 2014.
- MAT, R., SHARIFF, A., ZULKIFLI, A., RAHIM, M. e MAHAYUDIN, M., 2014, Using game engine for 3D terrain visualization of GIS data: A review, In *IOP Conference Series: Earth and Environmental Science*, 20(1), 012037.
- MILLER, C. e LAFLAMME, R., 1958,. *The Digital Terrain Model:- Theory & Application*, M.I.T. Photogrammetry Laboratory
- MILLER, G., 1986, The definition and rendering of terrain maps, *ACM SIGGRAPH Computer Graphics*, 20(4), pp. 39-48.
- MILLER, H., 1999, Potential contributions of spatial analysis to geographic information systems for transportation (GIS-T), *Geographical Analysis*, 31(4), pp. 373-399.

- MUSGRAVE, F. e MANDELBROT, B., 1991, The art of fractal landscapes, *IBM Journal of Research and Development*, 35(4), pp. 535-540.
- MUSGRAVE, F., KOLB, C. e MACE, R., 1989, The synthesis and rendering of eroded fractal terrains, *ACM SIGGRAPH Computer Graphics*, 23(3), pp. 41-50.
- NITSCHKE, M., 2008, *Video game spaces: image, play, and structure in 3D game worlds*, MIT Press.
- NITSCHKE, M., 2012, Growing Game Worlds, *Computer Games and New Media Cultures*, pp. 161-172.
- OPENSHAW, S., 1991, Developing appropriate spatial analysis methods for GIS, *Geographical Information Systems: principles and applications*, 1, pp. 389-402.
- PAGULAYAN, R., KEEKER, K., WIXON, D., ROMERO, R. e FULLER, T., 2003, User-centered design in games, *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, pp. 883-906.
- PARBERRY, I., 2013, Designer worlds: Procedural generation of infinite terrain from real-world elevation data, *Technical Report LARC 2013-02*.
- PARBERRY, I., 2014, Tobler's First Law of Geography, Self Similarity, and Perlin Noise: A Large Scale Analysis of Gradient Distribution in Southern Utah with Application to Procedural Terrain Generation, *Technical Report LARC 2014-04*.
- PATEL, A., 2010, Polygonal map generation for games, <http://www-cs-students.stanford.edu/~amitp/game-programming/polygon-map-generation/>, consultado a 22 de maio de 2014.
- PERLIN, K., 1985, An image synthesizer, *ACM Siggraph Computer Graphics*, 19(3), pp. 287-296.
- PERSSON, M., 2011, Terrain Generation, <http://n0tch.tumblr.com/post/4231184692/terrain-generation-part-1>, consultado a 6 de junho de 2014.
- PRUSINKIEWICZ, P. e HAMMEL, M., 1993, A fractal model of mountains and rivers. In *Canadian Information Processing Society: Graphics Interface*, 93, pp. 174-180.

- ROUSE III, R., 2010, *Game design: Theory and practice*, Wordware Pub.
- SALEN, K., e ZIMMERMAN, E., 2004, *Rules of play: Game design fundamentals*, MIT Press.
- SHEPHERD, I. e BLEASDALE-SHEPHERD, I., 2009, Videogames: the new GIS?, Middlesex University Research Repository
- SILVA, P. e COELHO, A., 2010, Procedural modeling of urban environments for digital games development, In *Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology*, pp. 112-113.
- SMELIK, R., DE KRAKER, K., TUTENEL, T., BIDARRA, R. e GROENEWEGEN, S., 2009, A survey of procedural methods for terrain modeling, pp. 25-34.
- SMITH, A. e MATEAS, M., 2011, Answer set programming for procedural content generation: A design space approach, *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3), pp. 187-200.
- SMITH, G., GAN, E., OTHENIN-GIRARD, A. e WHITEHEAD, J., 2011, PCG-based game design: enabling new play experiences through procedural content generation, In *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, p. 7.
- SQUIRE, K. e JENKINS, H., 2002, The art of contested spaces, In *Game on*, pp. 64-75.
- STOLE, M., 2011, A plasma fractal with python, <http://insolubleproblem.blogspot.pt/2011/01/plasma-fractal-with-python.html>, consultado a 30 de abril de 2014.
- SUTTON-SMITH, B., 1986, *Toys as Culture*, Gardner Press
- TAKATSUKI, Y., 2007, Cost headache for game developers, <http://news.bbc.co.uk/2/hi/business/7151961.stm>, consultado a 4 de agosto de 2014.
- TARBOTON, D., BRAS, R. e RODRIGUEZ-ITURBE, I., 1991, On the extraction of channel networks from digital elevation data, *Hydrological processes*, 5(1), pp. 81-100.



- TOGELIUS, J., YANNAKAKIS, G., STANLEY, K. e BROWNE, C., 2011, Search-based procedural content generation: A taxonomy and survey, *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3), pp. 172-186.
- TOMLIN, C. D, 2013, *GIS and cartographic modeling*, Esri Press.
- WEBSTER, A., 2010, Masterpiece: Final Fantasy VII, <http://arstechnica.com/gaming/2010/05/masterpiece-final-fantasy-vii/>, consultado a 16 de Setembro de 2014
- WEIBEL, R. e HELLER, M., 1993, *Digital terrain modeling*, Oxford University Press.
- WHITTAKER, R., 1975, *Communities and ecosystems*, 2<sup>a</sup> ed., McMillan.
- WOLF, M., 1997, Inventing space: Toward a taxonomy of on-and off-screen space in video games, *Film quarterly*, pp. 11-23.

## ANEXO 1 – CÓDIGO PYTHON DO SCRIPT PERLIN FBM

---

```
from arcpy import *
import os
from cgkit.noise import *
import random

mxd = mapping.MapDocument("CURRENT") #definição do documento em uso

#variáveis de input
dimensao=int(arcpy.GetParameter(0))
periodo=(arcpy.GetParameter(1))
oitavas=int(arcpy.GetParameter(2))
amp=int(arcpy.GetParameter(3))
fileName=arcpy.GetParameterAsText(4)

#lacunaridade e ganho por omissão
lacunarity=2.0
gain=0.5

#tamanho da célula por omissão
cellsize=25

#abre ficheiro para escrita nas especificações ASCII Grid
f=open(fileName, "w")

f.write('NCOLS %d\n'%dimensao)
f.write('NROWS %d\n'%dimensao)
f.write('XLLCORNER %d\n'% 0)
f.write('YLLCORNER %d\n'% 0)
f.write('CELLSIZE %d\n'% cellsize)
f.write('NODATA_VALUE -9999\n\n')

#escreve o valor da função fBm para cada célula
for x in range(dimensao):
    for y in range(dimensao):
        f.write('%s ' %str(amp*fBm((x/periodo,y/periodo,0),oitavas, lacunarity, gain)))
f.close()

#adiciona resultado à Data Frame em uso
df = mapping.ListDataFrames(mxd)[0]
result = MakeRasterLayer_management(fileName, "perlinfBm")
layer = result.getOutput(0)
mapping.AddLayer(df, layer, 'AUTO_ARRANGE')
```

## ANEXO 2 – CÓDIGO PYTHON DO SCRIPT DIAMOND SQUARE

```
#código adaptado de http://insolubleproblem.blogspot.pt/2011/01/plasma-fractal-with-
#python.html (Martin Stolle)

import random
import os
from arcpy import *

mxd = mapping.MapDocument("CURRENT") #definição do documento em uso
mapa=[] #inicializa a lista que vai registar x,y,valor

#variáveis de input
size=int(arcpy.GetParameter(0))
cellsize=int(arcpy.GetParameter(1))
roughness=int(arcpy.GetParameter(2))
fileName=arcpy.GetParameterAsText(3)

#função de deslocamento do ponto central
def diamondsquare(x, y, width, height, c1, c2, c3, c4):

    newWidth = round(width / 2) #posição do ponto central no eixo dos x
    newHeight = round(height / 2) #posição do ponto central no eixo dos y

    if (width>1 or height>1):
        #valor no ponto central
        midPoint = (c1 + c2 + c3 + c4) / 4 + width/size*roughness*random.uniform(-1,1)

#valor nos pontos médios das arestas
    edge1 = (c1 + c2) / 2
    edge2 = (c2 + c3) / 2
    edge3 = (c3 + c4) / 2
    edge4 = (c4 + c1) / 2

    #chama função para cada uma das 4 novas áreas
    diamondsquare(x, y, newWidth, newHeight, c1, edge1, midPoint, edge4)
    diamondsquare(x + newWidth, y, newWidth, newHeight, edge1, c2, edge2, midPoint)
    diamondsquare(x + newWidth, y + newHeight, newWidth, newHeight, midPoint, edge2, c3,
edge3)
    diamondsquare(x, y + newHeight, newWidth, newHeight, edge4, midPoint, edge3, c4)
    else:
        #escreve na variável mapa o valor na coordenada x,y
        c=(c1 + c2 + c3 + c4)/4
        mapa.append([x,y,c])

diamondsquare(0.0,0.0,size,size,0,0,0,0) #função chamada

sortedmap=sorted(mapa) #ordenados os valores na variável mapa

#abre ficheiro para escrita nas especificações ASCII Grid
f=open(fileName, 'wt')
f.write('NCOLS %d\n'%size)
f.write('NROWS %d\n'%size)
f.write('XLLCORNER %d\n'% 0)
f.write('YLLCORNER %d\n'% 0)
f.write('CELLSIZE %d\n'% cellsize)
f.write('NODAT_VALUE -9999\n\n')

#escreve o valor c para cada célula
for i in range(len(sortedmap)):
    f.write('%s ' %str(sortedmap[i][2]))
f.close()

#adiciona resultado à Data Frame em uso
df = mapping.ListDataFrames(mxd)[0]
result = MakeRasterLayer_management(fileName, "diamondSquare")
layer = result.getOutput(0)
mapping.AddLayer(df, layer, 'AUTO_ARRANGE')
```